*Manuscript
Click here to view linked References

LEGION-Based Image Segmentation by Means of Spiking Neural Networks Using Normalized Synaptic Weights Implemented on a Compact Scalable Neuromorphic Architecture

Giovanny Sanchez[a,*], Jordi Madrenas[b], Jordi Cosp-Vilella[b]

[a] *Instituto Politecnico Nacional, ESIME Culhuacan, Av. Santa Ana N 1000, Coyoacan, 04260, Distrito Federal, Mexico*
[b] *Dept. of Electronics Engineering, Universitat Politècnica de Catalunya, Jordi Girona, 1-3, edif. C4, 08034 Barcelona, Catalunya, Spain*

Abstract

LEGION (Locally Excitatory, Globally Inhibitory Oscillator Network) topology has demonstrated good capabilities in scene segmentation applications. However, the implementation of LEGION algorithm requires machines with high performance to process a set of complex differential equations limiting its use in practical real-time applications. Recently, several authors have proposed alternative methods based on Spiking Neural Networks (SNN) to create oscillatory neural networks with low computational complexity and highly feasible to be implemented on digital hardware to perform adaptive segmentation of images. Nevertheless, existing SNN with LEGION configuration focus on the membrane model leaving aside the behavior of the synapses although they play an important role in the synchronization of several segments by self-adapting their weights. In this work, we propose a SNN-LEGION configuration along with normalized weight of the synapses to self-adapt the SNN network to synchronize several segments of any size and shape at the same time. The proposed SNN-LEGION method involves a global inhibitor, which is in charge of performing the segmentation process between different objects with different sizes

*Corresponding author
Email address: giovas666@hotmail.com (Giovanny Sanchez)

and shapes on time. To validate the proposal, the SNN-LEGION method is implemented on an optimized scalable neuromorphic architecture. Our preliminary results demonstrate that the proposed normalization process of the synaptic weights along with the SNN-LEGION configuration keep the capacity of the LEGION network to separate the segments on time, which can be useful in video processing applications such as vision processing systems for mobile robots, offering lower computational complexity and area consumption compared with previously reported solutions.

Keywords: Spiking neural networks, LEGION, FPGA, SIMD architecture

1. Introduction

Over the last twenty years, image segmentation has become an attractive field due to its use in multiple practical applications, such as text recognition [1], face recognition [2], object recognition [3]. To perform image segmentation sucessfully, several authors have proposed a large number of segmentation algorithms, such as recurrent neural networks [3], spiking neural networks [4–6], Hidden Markov [1, 2], liquid state machine [7], LEGION [8]. In particular, several studies have demonstrated that LEGION algorithm [8] is a consistent approach in the development of practical applications, such as medical image segmentation [9, 10], segmentation in microscopy images [11], satellite image segmentation [12], among others. However, the LEGION algorithm achieves high computational capabilities in image segmentation by employing a set of complex differential equations. As a consequence, LEGION algorithm demands high computational cost restricting its practical use in video processing applications. One potential solution to increase the processing speed can be found in parallel computing since it has become increasingly important for efficient development of image processing applications. From an engineering perspective, LEGION algorithm can be implemented in parallel architectures since its structure is highly parallel, so that, high operation speeds can be guaranteed. On the other hand, oscillatory spiking neural networks (SNN) have emerged to be

applied in procedures of image segmentation since these new algorithms exhibit similar principles of operation of LEGION network and offer lower complexity [13–15]. Therefore, the development of new schemes for efficient image segmentation could then take advantage of the positive aspects of LEGION network and the spiking neural networks. In addition, there is still the need to develop suitable hardware architectures to process these new approaches at high processing speeds. In particular, researchers in electronics and computer science have made intense efforts to create efficient computer systems for image segmentation. However, there are many challenges to be solved, especially in the development of highly parallel architectures. Some of these SNN implementations have been carried out in general purpose computers due to their great flexibility in terms of programmability. However, memory accesses limit the processing speed and increase power consumption. In contrast to software implementations, analog implementations offer interesting features in terms of power-area consumption and processing speed by paying a penalty in programmability. One of the first works developed in VLSI for scene segmentation was proposed by Cosp et al. [16]. This work presents the development of a neuromorphic architecture with a minimum power consumption to be used in portable systems. Nowadays, FPGA devices are considered as an important development tool since they offer advanced memory systems and communication systems which allows to create highly parallel computing systems. Several authors have implemented their SNN models with LEGION configuration in FPGA. One of these works was proposed by Cheung et al. [14]. The authors proposed a system to simulate a network with 800 Izhikevich neurons in an FPGA. On the other hand, Torres et al. [13] propose a massively distributed digital implementation to perform the segmentation of 16x16 images with a connectivity of 8 neighboring neurons. Analysing previous works, we observed that the oscillatory spiking neural networks employ large number of synapses to carry out the process of synchronization and de-synchronization of the spiking neural oscillators in the network. In particular, they use several synapses to keep low the computational complexity at the cost of increasing the area consumption. This aspect becomes critical when

large-scale spiking neural networks need to be implemented in advanced hardware architectures since the implementation of synapses is the most demanding factor in terms of area consumption. In this brief, we present a normalization process of the synaptic potentials along with the SNN-LEGION configuration to carry out simultaneous discrimination of different segments by guaranteeing lower computational complexity. Its implementation on an optimized scalable parallel architecture requires lower number of synapses compared with existing approaches.

2. A brief introduction to the LEGION algorithm

Experimental studies have demonstrated that oscillations of neurons in the visual cortex play an important role in the image segmentation, i.e., the visual cortex employs these oscillations to segment objects in a visual scene [8, 17, 18]. Neurons syncronize their firing patterns to achieve such segmentation. So far, several algorithms have been proposed inspired by these neural phenomena. One of the first proposal was introduced by Wang and Terman [8], who presented the locally excitatory globally inhibitory oscillator network (LEGION) algorithm. This algorithm is composed of a two-dimensional (2-D) array of relaxation oscillators in which the local synchronism is carried out by oscillators locally connected with positive coupling (excitatory synapses) and desynchronization between synchronized groups of local oscillators is performed by a global oscillator by means of negatively coupled (inhibitory synapses), as shown in Fig. 1.

In binary image segmentation, a set of local oscillators are grouped to characterize an object in which each local oscillator is linked to a pixel. Therefore, an object is defined as a group of pixels that share the same characteristic (black or white). These characteristics are linked to the status of the local oscillators, i.e., when oscillator states are represented as an output image, black pixels of this image represent the high state (active) of the local oscillators while white pixels point out the low state (silent) of the local oscillators. To achieve image

segmentation, each group of local oscillators must be synchronized in a single phase of oscillation to encode an object. Once the groups get formed, they are desynchronized from other groups of local oscillator by means of inhibition from the global oscillator. Most of the existing approaches require four connections to have equal overall weight of dynamic connections to guarantee synchronization between objects. Here, the size and shape of the object defines the number of oscillators to be used. Therefore, not all oscillators are excited, i.e., some local oscillators are stimulated either by 4, 3, 2 or 1 nearest neighbors. A potential solution to ensure that each oscillator contains the same overall weight of dynamic connections from its neighborhood can be found in the use of weight normalization process. However, a set of complex differential equations describe the oscillators and their normalization process [8, 19]. This factor limits its use in practical video segmentation applications since its implementation demands high power computing. In this brief, we propose a new alternative method based on SNN-LEGION along with a synaptic weight normalization to guarantee oscillator synchronization whether they are stimulated by one, two, three or four nearest neighbors requiring less power computing when compared with conventional LEGION. Besides, we proposed a customized architecture to perform the proposed SNN-LEGION at high processing speeds and it requires the minimum amount of area compared with existing hardware architectures. The combination of these two approaches (SNN-LEGION algorithm and hardware architecture) potentially allow the development of practical video segmentation applications.

3. Spiking Neural Network with LEGION configuration

The proposed alternative method includes a synaptic weight normalization process to self-adapt the SNN network to synchronize several segments of any size and shape at the same time. Here, the global oscillator produces an inhibition to all local oscillators to desynchronize different groups of local oscillators (segments). This process is known as a segmentation process. Each local ex-

citatory oscillator is connected to its four nearest neighbors, as shown in Fig. 1. Here, local excitatory neurons and the global inhibitory neuron are modelled as leaky integrate-and-fire (LIF) neurons and the normalization process of the weights is carried out in the synapses of the local excitatory neurons. We propose a normalization process of the synaptic weights inspired by the long-term potentiation (LTP) and long-term depression (LTD), which produce an increase and decrease in the synaptic strength, respectively, by processing recent patterns of activity [20]. Inspired by this phenomena, we include the modified LTP and LTD rules in the synapse model to perform the normalization by self-adapting the synaptic weights. Therefore, each local excitatory neuron can be stimulated by any number, from one to four, of their nearest neighbors to achieve synchronization.

The membrane potential dynamics is described as follows:

$$V(t+1) = V_{rest} + B(t) + (1 - S_i(t))\left((V(t) - V_{rest})\, k_{mem}\right) + \sum_j \omega_{ji}(t) \qquad (1)$$

where $V(t+1)$ is the membrane potential of neuron, $B(t)$ is the background activity noise, V_{rest} is the value of the resting potential, $k_{mem} = e^{\left(-\frac{\Delta t}{\tau_{mem}}\right)}$ is the time constant associated with the leakage current, $S_i(t)$ is the post-synaptic spike, and finally, $\omega_{ji}(t)$ is the pre-synaptic weight. Here, the post-synaptic spike $S_i(t)$ is a function that depends on the membrane potential $V(t)$ and the threshold potential $V(\theta)$, i.e., if the membrane potential $V(t)$ crosses the threshold potential $V(\theta)$, the neuron generates a post-synaptic spike ($S_i(t) = H(V(t) - V(\theta))$, where H represents the Heaviside function.

3.1. The proposed normalization mechanism for local excitatory neurons

The synchronization of a distinct group of local excitatory neurons, which represents an object, occurs when local excitatory neurons receive excitatory potential from their four nearest neighbors, except on boundaries where the nearest neighbors are limited to 3 or 2, as shown in Fig. 1. The proposed normalization

mechanism carried out in the synaptic weights is formulated mathematically, as follows:

$$\omega_{ji}(t+1) = S_j(t)P(L_j(t)) \tag{2}$$

where $S_j(t)$ is the pre-synaptic spike, P defines either the inhibitory or excitatory synaptic potential. Here, the inhibitory synaptic potential is fixed ($P(t)$ = -4 mV), whereas, the excitatory synaptic potential is variable since it depends on synaptic variable $L_j(t)$, which can be either increased or decreased depending on the number and the inter-spike period of the pre-synaptic spikes $S_j(t)$ and is defined as follows:

$$L_j(t+1) = L_j(t)k_{act} + S_j(t)P_{min} \tag{3}$$

Equation 3 indicates that if the excitatory synapse of a local excitatory neuron receives a pre-synaptic spike $S_j(t)$ generated by its nearest neighbor, the synaptic variable $L_j(t)$ is increased by a value defined as P_{min}. Otherwise, the synaptic variable $L_j(t)$ decreases exponentially to zero as a function of a time constant $k_{act} = e^{\left(-\frac{\Delta t}{\tau_{syn}}\right)}$, but it can be increased in case of receiving again pre-synaptic spikes $S_j(t)$. Hence, the increase or decrease of the synaptic potential $P(t)$ is a function of the value of $L_j(t)$, as shown in Fig. 2. The normalization of synaptic potentials $P(t)$ is performed when for a given j $L_j(t) = 0$, indicates that synapse j is inactive. Therefore, the active synapses increase or decrease their synaptic potential $P(t)$ to ensure that each local excitatory neuron has the same overall weight of dynamic connections from its neighborhood, as follows:

$$P(L_j(t) \neq 0) = \frac{(V_{rest} - V(\theta))}{\sum P(L_j(t) \neq 0)} \tag{4}$$

Equation 4 indicates that the increase or decrease of the synaptic potentials $P(t)$ of the four synapses is carried out when any of the synapses of a local excitatory neuron becomes inactive ($L_j(t) = 0$). Here, each single local excitatory

neuron generates a post-synaptic spike in the case that the membrane potential $V(t)$ is equal to the threshold potential $V(\theta)$. Therefore, membrane potential $V(t)$ can reach the threshold potential $V(\theta)$ only if the membrane potential $V(t)$ receives enough stimulation by means of synaptic potentials $P(t)$ according to equations 1 and 2. To ensure such stimulation, each synaptic potential $P(t)$ is determined by the difference between the resting potential V_{rest} and threshold potential $V(\theta)$. The result of this difference is divided by the number of active synapses, as formulated in equation 4.

4. Neuromorphic architecture overview

Since one of our objectives is to create a prototype to be used in vision processing systems for mobile robots in which real-time processing is demanded and area resources are limited, we optimize an existing multi-model SNN architecture called SNAVA [21] to simulate the proposed SNN-LEGION configuration at high processing speeds by requiring low area consumption. It should be mentioned that the existing SNAVA architecture was designed as a generic multi-model SNN platform to simulate large-scale SNN networks paying a penalty in terms of area consumption and speed processing. In SNAVA architecture, the simulation of large number of neurons can be done using the virtualization concept, i.e., each processor simulates a large number of neurons sequentially. This factor reduces significantly the performance of the SNAVA architecture by simulating several neurons sequentially. Here, we have made significant changes to the existing SNAVA architecture to create a compact and high-speed neuromorphic architecture. Specifically, we remove and redesign some components of the SNAVA architecture used for supporting the time-multiplexing of neural computation. Therefore, the optimized neuromorphic architecture simulate a single neuron per each processor. This change have allowed us to reduce significantly the processing time and the area consumption.

The proposed scalable neuromorphic architecture is mainly composed of three modules (scalable PE array, AER module and system manager), as shown

in Fig. 3. Essentially, the scalable array is composed of Processing Elements (PE), which are the basic building blocks of the system. The AER module is in charge of establishing the communication between PEs within the neuromorphic architecture and is composed of an AER address generator and an AER controller. Finally, the system manager involves a block RAM (BRAM) memory, a sequencer and a configuration unit. In addition to the components mentioned above, there is an external CPU used to access the chip for response analysis and initial configuration.

The details of these three modules are given as follows:

1. Scalable array of PEs. Each PE simulate a single neuron using a 16-bit synaptic BRAM, two 16-bit register banks, a 15-bit content-addressable memory (CAM), a 1-bit spike register and a 16-bit ALU, as shown in Fig. 4. We intend to maximize the intrinsic parallelism of the proposed scalable neuromorphic architecture by using distributed memory. Therefore, each 16-bit synaptic BRAM stores/loads the synaptic parameters expending two clock cycles and all synaptic BRAMs are updated in parallel. The register banks, which are called active and shadow registers, are used to store/load neural parameters expending one clock cycle. A significant improvement is achieved by storing neural parameters in a shadow registers. In SNAVA, the neural parameters are stored in a BRAM block, where a set of memory positions correspond to multiple neural parameters of a specific virtual neuron. Therefore, each PE expends several clock cycles to read/load several neural parameters. In the current version, each PE can access to multiple shadow registers to read/load neural parameters expending a single clock cycle since the ALU is connected to shadow register bank directly, as shown in Fig 5. The use of these storage elements distributed in each PE have allowed to improve the processing speed since the memory data transfer represent a big bottleneck in modern parallel architectures. In addition, each PE contains a CAM used to store the address of the synapses, i.e., the synapse connectivity between PEs. The

spike register indicates of a pre-synaptic spike arrives at specific neuron. Usually, the ALU operates on active registers where register 0 is always referred to as the accumulator. The shadow register serves as a temporary storage for the active registers by providing space for SNN algorithms with a large number of neural parameters. Data move operations are possible between shadow and active registers either as single or bulk. Finally, the ALU was designed to support the required arithmetic operations, such as two's complement subtraction and addition, shifting, xor, and, or, negation and multiplication, to simulate the proposed SNN-LEGION network efficiently. This has allowed us to achieve lower area consumption compared with SNAVA.

2. System manager. The system manager is composed of a block RAM (BRAM), a sequencer, a configuration unit. The sequencer is responsible for controlling the program flow by decoding and fetching the instructions and constants stored in a BRAM memory and broadcasting the instructions to be executed by the scalable PE array. Finally, the configuration unit manages the configuration of different components of the neuromorphic architecture, such as AER module, sequencer, PE array, etc. It also sets registers to enable the debugging capabilities of the neuromorphic architecture, such as debugging step by step, setting the clock mode, enabling and disabling its components.

3. AER module. The AER module is composed of an AER controller, an AER address generator to perform the spike distribution among PEs [22]. The AER address generator reads the spikes produced by the multiprocessor array and distributes them within the same multiprocessor via AER bus. The AER controller sends all post-synaptic spikes produced by the PEs and each CAM receives them and compares them with a set of destination PE ID's. In case of matching both directions (source and destination), the CAM writes the corresponding pre-synaptic spikes into the spike register.

The functional operation of the configurable neuromorphic architecture was designed to work in two operational phases, which are the data processing (phase 1) and the spike distribution (phase 2), as shown in Fig. 6. Therefore, the simulation of SNN models are executed in these two periodic phases. In the processing phase, the neural and synaptic parameters are computed. Then, in the distribution phase, the post-synaptic spikes generated by the neurons in phase 1 are distributed by the AER module through the SNN network. The spike distribution is carried out by the synchronous AER protocol proposed in [23] to avoid overhead connection when a large-scale SNN models are implemented.

5. Implementation and performance analysis

The proposed 16-bit fixed-point neuromorphic architecture is implemented on a Spartan 3-XC3S5000 FPGA running at 50 MHz. It can support an array of 6x6 16-bit PEs with 30 synapses per PE. The implementation of whole configurable neuromorphic architecture requires 53% of LUTs and 10% of registers in this device. In this work, we enabled three sets of arbitrary objects with different sizes and shapes using 30 local excitatory neurons and a global inhibitory neuron according to the network configuration of Fig. 1. For different simulations, several configurations have been tried. The first group of local excitatory neurons define two triangles, the second group of local excitatory neurons define two trapezoids and the third group of local excitatory neurons includes a inverted letter L, a square, a small column and half a trapezoid, as shown in Figures 7 8 9, respectively.

Table 1: Synaptic-neural parameters

Synaptic-Neural Variable	Symbol	Excitatory	Inhibitory
Resting potential	V_{rest}	–300 mV	–300 mV
Threshold potential	$V(\theta)$	–280 mV	–280 mV
Synaptic potential	$P(t)$	6 mV	–4 mV
Synaptic variable (initial)	$L_j(t)$	8191	

Table 1 shows the synaptic-neural parameters used to segment the three groups of objects mentioned above. In this experiment, τ_{mem} = 50 ms and Δt = 0.5 ms, τ_{syn} = 10 ms and Δt = 0.1 ms. The background activity noise $B(t)$ is an injected random excitatory pre-synaptic potential with zero mean and 30 mV standard deviation that produces an increment of the membrane potential $V(t+1)$. The global inhibitory neuron fires post-synaptic spikes due to noise at an average rate of 3 spikes/s. According to values of Table 1, the initial value of the synaptic potential $P(t)$ is 6 mV, so that, each pre-synaptic spike $S_j(t)$ increases 6 mV the membrane potential $V(t+1)$ only when its four synapses are active. Therefore, the membrane potential of each local excitatory neuron must receive 24 mV, which is the potential associated with the pre-synaptic weight $\omega_{ji}(t)$, by means of its four synapses to reach the threshold potential $V(\theta)$, and thus generates a post-synaptic spike $S_i(t)$. However, in some cases not all synapses are active, thus, we proposed the normalization of synaptic weights by self-adapting the synaptic potentials $P(t)$ to guarantee that each local excitatory neuron contains the same overall weight of dynamic connections from its neighborhood, as shown in Fig. 10. Here, a synapse is considered active as long as the value of the synaptic variable $L_j(t)$ is different from 0, otherwise it is considered inactive. Hence, the normalization of the synaptic weights is carried out whenever the value of the variable $L_j(t)$ of any synapse is equal to zero, thereby the synaptic potentials $P(t)$ are distributed in such a way that the sum of these synaptic potentials $P(t)$ is equal to 24 mV. Here, the synaptic potential $P(t)$ of any synapse can take any of the following values: 0, 6, 8, 12 and 24 mV (see Fig. 10) in function of the amount of excitation of its neighborhood.

Once the synaptic-neural values of the parameters of the SNN-LEGION network were defined, equations 1, 2 and 3 were programmed in assembly language to achieve maximum efficiency in terms of processing speed. The execution loop, which contains the subroutines to execute the data processing and spike distribution phases, is shown in Fig. 11. Here, neural constants (V_{rest}, $V(\theta)$) and synaptic constants (k_{mem}, P_{min}, k_{act}) are stored in a single BRAM since these constants are common for all the PEs, as shown in Fig. 3. The initialization

process consist of a single subroutine called "Neuron load". This subroutine is in charge of loading the initial value of neural variables, such as the membrane potential $V(t)$, the pre-synaptic weight $\omega_{ji}(t)$, and the background activity noise $B(t)$ into their respective shadow registers, as shown in Fig. 12. Basically, a synaptic loop and a set of subroutines are dedicated to perform the data processing phase. Specifically, the synapse loop is dedicated to compute the variable synaptic parameters ($P(t)$, $L_j(t)$), and a neural parameter $\omega_{ji}(t)$). Therefore, the number of times the synapse loop is executed depends on the number of synapses. This loop is composed of 6 subroutines, such as "synapse load", "synaptic deactivation", "synaptic activation", "synaptic potential distribution", "synaptic variable" and "synaptic weight".

1. The subroutine "synapse load" loads the synaptic variables ($P(t)$, $L_j(t)$) from the synaptic BRAM to active registers.
2. The subroutine "synaptic deactivation" serves to indicate which synapse is inactive.
3. The subroutine "synaptic activation" indicates which synapse is active.
4. Once the active or inactive synapses are detected, the subroutine "synaptic potential distribution" performs the normalization process, i,e., it calculates the synaptic potential $P(t)$ per each synapse, as shown in equation 4.
5. The subroutine "synaptic variable" updates the value of $L_j(t)$ by executing the equation 3.
6. The subroutine "synaptic weight" is dedicated to compute the synaptic weights (see equation 2).

On the other hand, we program six subroutines, such as "neuron membrane value", "spike update", "noise", "refractory period", "spikes enable" and "neuron save" to calculate the neural parameters.

1. The subroutine "neuron membrane value" calculates the membrane potential $V(t)$ as defined in equation 1.

2. The subroutine "spike update" performs a comparison between the value of the membrane potential $V(t)$ and the value of the threshold potential $V(\theta)$. If the value of membrane potential $V(t)$ is greater or equal to threshold potential $V(\theta)$, neuron generates a post-synaptic spike S_i.
3. The subroutine "noise" generates the background activity noise $B(t)$.
4. The subroutine "refractory period" disables neurons for a specific slot of time, i.e., once the neuron have generated a post-synaptic spike S_i, the neuron cannot generate a second a post-synaptic spike S_i, intermediately.
5. The subroutine "spikes enable" indicates the post-synaptic spikes, which were generated by the neurons.
6. The subroutine "neuron save" is in charge of transferring the neural values $V(t+1)$ and $B(t)$ from the active registers to shadow registers.

Finally, the subroutine "STOP" indicates that the spike distribution can be executed by means of AER module. Once the "STOP" is executed, the data processing phase is executed again. Here, every subroutine requires a certain number of clock cycles to perform a specific part of the algorithm. The number of clock cycles per subroutine were measured and translated into a mathematical equation as a function of constants K and variable S, where variable S define the number of synapses to be implemented in the optimized scalable neuromorphic architecture. Equation 5 calculates the total number of clock cycles required to execute the assembler program (see Fig. 11).

$$N_T = K + 34 \cdot S \tag{5}$$

Replacing constants and variable ($K = 33$ and $S = 4$) in equation 5, the total number of clock cycles N_T is 169. To obtain the execution time, the total number of clock cycles is multiplied by the system clock period (20 ns), therefore, the execution time is 3.3 μs to carry out the calculation of the neuronal and synaptic parameters in a single simulation cycle.

6. Results

This section presents the results of the implementation of the proposed SNN-LEGION network along with normalization weight process in the configurable neuromorphic architecture.

Before presenting the results, we show in a general way how the proposed SNN-LEGION network performs binary image segmentation of several objects with different shapes and sizes by employing the minimum number of synapses. To achieve binary image segmentation, the proposed SNN-LEGION network executes three tasks, simultaneously, as follows:

1. Local excitatory neurons, which represent an object, are stimulated by receiving pre-synaptic spikes $S_j(t)$ at each simulation step. Therefore, the membrane potential of these neurons is increased by 6 mV synaptic potential. In addition, these neurons receive stimulus from the background activity noise $B(t)$ to generate approximately 3 spikes/s. During the initial simulation steps, local excitatory neurons fire spikes randomly because of these two stimuli.
2. After some simulations steps, locally excitatory coupled neurons become synchronized by self-adapting the synaptic weights of the synapses to represent segments (objects). Here, the proposed normalization process helps to improve the synchronization between segments, i.e., each local neuron requires four connections to have equal overall weight of dynamic connections to guarantee synchronization between objects.
3. During the whole segmentation process, a global inhibitor generates post-synaptic spikes whenever any neuron in the network becomes active. As a consequence, these spikes decrease the membrane potential of all local excitatory neurons, i.e., if a neuron belongs to the object whose neurons are active and have fired the global inhibitor, it will receive an inhibitory synaptic potential ($P(t)$ = -4 mV) from the global inhibitor and the excitatory synaptic potentials from its local connections, as described in Eq. 2. The overall result will be a positive synaptic potential since the ma-

jority of the local connections are excitatory, i.e., the sum of excitatory potentials is always greater than the inhibitory synaptic potential for this neuron. Therefore, this will maintain the synchronization with the other firing neurons. However, if the neuron does not belong to the firing object, it will only receive the inhibitory potential, which is negative, thus, delaying the firing of the neuron and desynchronizing it from the active object that fired first.

Next, we present the results, which have been verified by means of a Graphical User Interface. We have developed this GUI to monitor the network and graphically display the neural parameters for easy perception and analysis on real-time [21]. According to previous description, the binary segmentation image process begins when the AER controller (see Fig. 3) writes the pre-synaptic spikes $S_j(t)$ in their respective SRAM memory positions that correspond to the synapses of local excitatory neurons (see Fig. 7). In this way, the neuron is stimulated by receiving pre-synaptic spikes $S_j(t)$. On the other hand, we enabled a linear-feedback shift register (LFSR) to generate the background activity noise $B(t)$, which produces a stimulus to all the neurons (excitatory and inhibitory) in the network, resulting in all neurons generating approximately 3 spikes/s.

The experiment consists of the binary image segmentation of the three sets of objects, as shown in Figures 7, 8 and 9. In this experiment each set of objects were introduced to the SNN-LEGION network one by one manually, as follows:

1. *Image segmentation of triangles.* As it can be observed in Figures 13 and 14, the AER module generates and sends pre-synaptic spikes $S_j(t)$, which represent the triangles, to the local excitatory neurons, during the first 1,240 simulation cycles. From simulation cycle number 45 to 445, local excitatory neurons fire spikes randomly due to the activation of background activity noise $B(t)$. The locally excitatory coupled neurons become synchronized when the synaptic normalization is done. This occurs at simulation cycle number 1040. The identification of the first triangle was carried out approximately at the simulation cycle number 1,020, i.e.,

the proposed SNN-LEGION network segments the binary image in 3.4 ms.

2. *Image segmentation of trapezoids*. Once the triangles were segmented satisfactorily, we introduced two trapezoids (see Fig. 8) to be segmented by the proposed SNN-LEGION configuration. Figure 15 shows the neural activity of the local excitatory neurons from simulation cycle 1800 to 2120. As can be observed, the local excitatory neurons generate spikes randomly because these neurons self-adapt their synaptic weights for a second time in order to segment the trapezoids 1 and 2. At simulation cycle 3,000, trapezoids 1 and 2 were segmented, as shown in Fig. 16. Therefore, the proposed SNN-LEGION network segments the trapezoids in 3.96 ms approximately.

3. *Image segmentation of four objects* (*one letter L*, *one square*, *one small column and one trapezoid*). We introduced a set of of objects (see Fig. 9) to the proposed SNN-LEGION configuration once the trapezoids were segmented properly. From the simulation cycle number 3,880 to the simulation cycle number 4,280, the local excitatory neurons perform the normalization process to self-adapt their weights for a third time, as shown in Fig. 17. Figure 18 clearly shows the segmentation of the objects (one letter L, one square, one small column and one trapezoid) at 6,620 simulation cycle. On this occasion, the proposed SNN-LEGION network spends 9.042 ms to perform the segmentation of four objects with different size and shape.

In general terms, the proposed SNN-LEGION network has demonstrated good capabilities in binary image segmentation by self-adapting the synaptic weights independently that local excitatory neurons are stimulated either by one, two, three or four of their synapses in contrast to the existing proposals in which their neurons must necessarily be stimulated by their four synapses. Here, we demonstrate that the proposed SNN-LEGION algorithm performs image segmentation at high processing speeds (in 3.96 ms approximately). This potentially allows its use in practical video processing applications since at 30

frames per second, i.e., a single video frame (image) lasts 33 ms.

7. Discussion

The existing approaches contemplate a LEGION configuration in which each local excitatory neuron has 8 synapses to guarantee the same overall weight of dynamic connections from its neighborhood since the size and shape of each segment is different from other segments [13, 15, 24]. K. Chen et al. [19] proposed a method for image segmentation based on LEGION with normalization process. To achieve the normalization of the weights, the authors proposed a network of oscillators in which each local oscillator has 8 connections. To adapt weights, the authors use two types of connections (dynamic and fixed). Here, dynamic connectivity is used to specify the instantaneous relationship between two adjacent oscillators during weight normalization. The authors have reported satisfactory results in image segmentation. However, the dynamic connections are modelled by means of logarithmic functions which represent large computational cost. As can be observed in previous works, most of the existing solutions employ 8 synapses to process in terms of size and shape while our proposal uses 4 synapses with dynamic adaptation of the synaptic potentials to process objects with different shape and size. Nowadays, supporting efficiently large amount of synapses in advanced neuromorphic architectures is still a challenge. Specifically, in the most significant work, the authors have implemented the synapses of the SNN-LEGION network by means of physical connections achieving higher processing speeds and expending lower area when compared with our implementation, as shown in Table 2. As can be observed, we made a consistent comparison between our proposal and existing approaches [13, 15, 24] in terms of processing speed and area resources by implementing the local excitatory neurons according to the reported results in [13, 15, 24]. However, some authors have proposed digital architectures, which has not been implemented in digital embedded devices, to simulate SNN-LEGION algorithms. Therefore, its performace has not been validated [24]. As can be observed in Table 2, the

existing approach [15] is approximately thousand times faster than the scalable neuromorphic architecture. Evidently, our proposal expends higher number of clocks since it includes normalization of the weights in contrast to other proposals, in which the synapses are considered as simple connections. This factor limits the image segmentation capabilities of other approaches by processing objects with regular size and shape. However, the optimazed scalable neuromorphic architecture requires approximately three times fewer registers than the configurable neuromorphic architectures do [13, 15]. In addition, it requires approximately half of LUTs. Besides, the existing architectures have critical paths and routing congestion, which affect severely their performance by supporting larger image sizes, as stated in [15]. As a consequence, these factors limit the scalability of their approaches. This is reflected in the fact that the existing approaches can support a reduced number of neurons using FPGAs with better area resources. Hence, an interesting feature of the proposed neuromorphic architecture is linked to its scalability feature, i.e., the PE array size can be increased easily without paying a penalty in terms of critical paths. Evidently, the configurable neuromorphic architecture is implemented in a low-cost FPGA in order to be used in vision processing systems for mobile robots in near future, where the area resources are limited.

In summary, our results have demonstrated that a local excitatory neuron can be stimulated by one of its nearest neighbor (worst case) to achieve the synchronization. In addition, the use of this method have allowed us to perform the image segmentation requiring the minimum number of synapses compared with existing solutions, as shown in Table 2.

8. Conclusions

This brief presents an alternative method for binary image segmentation based on spiking neural networks with LEGION configuration along with normalized synaptic weights to self-adapt the network. The proposed method considers the binary image segmentation since some authors have demonstrated

Table 2: Comparison between the existing approaches [13, 15] and the configurable neuromorphic architecture in terms of hardware elements by implementing a 6x5 LEGION network.

Approach	Device	Area	Time	Number of synapses	Normalization
This work	Spartan 3-XC3S5000	1,224 LUTs, 213 Registers	3.4 ms - 3.9 ms	120	YES
Girau *et al.* [15]	Virtex II XC2V6000-4FF1517	2,010 LUTs, 660 Registers	0.003 ms	240	NO
Torres *et al.* [13]	Virtex XC2V1500FF896-4	1,920 LUTs, 652 Registers	NR	240	NO
Fernandes *et al.* [24]	Max+plusII (Software of Altera)	NR	NR	240	NO

that the implementation of their alternative methods for binary image segmentation demands low area consumption [15]. As consequence, their alternative methods have been implemented in the current embedded devices to validate their efficiency. Here, the proposed method has allowed us to segment objects of any size and shape using a reduced number of synapses compared with other existing proposals. As a proof of concept, the segmentation of images has been experimentally demonstrated using SNN-LEGION connectivity along with normalized synaptic weights. The proposed method has allowed us to reduce the computational complexity of the original oscillator model [8] and the connections with normalization process [19], respectively. Preliminary results show that the optimized neuromorphic architecture represents an attractive alternative for applications that involve image segmentation by exhibiting great flexibility for practical implementations. Obviously, the current prototype implements a reduced number of neurons since part of the future work is to use the proposed method in vision processing systems for mobile robots. In addition, the proposed architecture will be translated into a customized VLSI design, which potentially allows a massive integration of spiking neurons to process high resolution images [25, 26].

Acknowledgments

The authors would like to thank the Consejo Nacional de Ciencia y Tecnologia (CONACyT) and the IPN for the financial support to realize this work. This work was also supported in part by the Spanish Ministry of Science and Innovation and the European Social Fund (ESF) under Projects TEC2011-27047 and TEC2015-67278-R.

References

[1] P. P. Roy, A. K. Bhunia, U. Pal, Date-field retrieval in scene image and video frames using text enhancement and shape coding, Neurocomputing 274 (2018) 37–49.

[2] F. Samaria, S. Young, Hmm-based architecture for face identification, Image and vision computing 12 (8) (1994) 537–543.

[3] B. Shuai, Z. Zuo, B. Wang, G. Wang, Scene segmentation with dag-recurrent neural networks, IEEE transactions on pattern analysis and machine intelligence 40 (6) (2018) 1480–1493.

[4] Q. Wu, M. McGinnity, L. Maguire, A. Belatreche, B. Glackin, Edge detection based on spiking neural network model, in: International Conference on Intelligent Computing, Springer, 2007, pp. 26–34.

[5] B. Meftah, M. Debakla, M. Zaagane, A. Benyettou, O. Lezoray, Spiking neuron network for image segmentation, in: Maghrebian Conference on Information Technologies, 2008, pp. 13–17.

[6] X.-S. Yang, Artificial intelligence, evolutionary computing and metaheuristics: in the footsteps of Alan Turing, Vol. 427, Springer, 2012.

[7] A. Polepalli, N. Soures, D. Kudithipudi, Digital neuromorphic design of a liquid state machine for real-time processing, in: Rebooting Computing (ICRC), IEEE International Conference on, IEEE, 2016, pp. 1–8.

[8] D. Terman, D. Wang, Global competition and local cooperation in a network of neural oscillators, Physica D: Nonlinear Phenomena 81 (1-2) (1995) 148–176.

[9] N. Shareef, D. L. Wang, R. Yagel, Segmentation of medical images using legion, IEEE transactions on medical imaging 18 (1) (1999) 74–91.

[10] G. Sivaradje, D. Saraswady, P. Dananjayan, Legionbased segmentation of magnetic resonance images of the brain, IETE journal of research 48 (3-4) (2002) 311–315.

[11] D. Fernandes, P. O. A. Navaux, P. F. P. Fichtner, Segmentation of tem images using oscillatory neural networks, in: Computer Graphics and Image

Processing, 2001 Proceedings of XIV Brazilian Symposium on, IEEE, 2001, pp. 289–296.

[12] J. Yuan, D. Wang, B. Wu, L. Yan, R. Li, Legion-based automatic road extraction from satellite imagery, IEEE transactions on geoscience and remote sensing 49 (11) (2011) 4528–4538.

[13] C. Torres-Huitzil, B. Girau, Massively distributed digital implementation of a spiking neural network for image segmentation on fpga, Neural Information Processing-Letters and Reviews 10 (4-6) (2006) 105–114.

[14] K. Cheung, S. R. Schultz, P. H. Leong, A parallel spiking neural network simulator, in: Field-Programmable Technology, 2009. FPT 2009. International Conference on, IEEE, 2009, pp. 247–254.

[15] B. Girau, C. Torres-Huitzil, Massively distributed digital implementation of an integrate-and-fire legion network for visual scene segmentation, Neurocomputing 70 (7-9) (2007) 1186–1197.

[16] J. Cosp, J. Madrenas, J. Moreno, J. Cabestany, Analog vlsi implementation of a relaxation oscillator for neuromorphic networks, in: Neuromorphic Systems: Engineering Silicon from Neurobiology, World Scientific, 1998, pp. 197–208.

[17] D. Wang, D. Terman, Image segmentation based on oscillatory correlation, Neural Computation 9 (4) (1997) 805–836.

[18] S. R. Campbell, D. L. Wang, C. Jayaprakash, Synchrony and desynchrony in integrate-and-fire oscillators, Neural computation 11 (7) (1999) 1595–1619.

[19] K. Chen, D. Wang, X. Liu, Weight adaptation and oscillatory correlation for image segmentation, IEEE Transactions on Neural Networks 11 (5) (2000) 1106–1123.

[20] J. Iglesias, J. Eriksson, F. Grize, M. Tomassini, A. E. Villa, Dynamics of pruning in simulated large-scale spiking neural networks, Biosystems 79 (1-3) (2005) 11–20.

[21] A. Sripad, G. Sanchez, M. Zapata, V. Pirrone, T. Dorta, S. Cambria, A. Marti, K. Krishnamourthy, J. Madrenas, Snavaa real-time multi-fpga multi-model spiking neural network simulation architecture, Neural Networks 97 (2018) 28–45.

[22] T. Dorta, M. Zapata, J. Madrenas, G. Sanchez, Aer-srt: Scalable spike distribution by means of synchronous serial ring topology address event representation, Neurocomputing 171 (2016) 1684–1690.

[23] A. Ferscha, S. K. Tripathi, Parallel and distributed simulation of discrete event systems, Tech. rep. (1998).

[24] D. Fernandes, J. P. Stedile, P. O. A. Navaux, Architecture of oscillatory neural network for image segmentation, in: Computer Architecture and High Performance Computing, 2002. Proceedings. 14th Symposium on, IEEE, 2002, pp. 29–36.

[25] J. Cosp, J. Madrenas, D. Fernández, Design and basic blocks of a neuromorphic vlsi analogue vision system, Neurocomputing 69 (16-18) (2006) 1962–1970.

[26] J. Cosp, S. Binczak, J. Madrenas, D. Fernández, Realistic model of compact vlsi fitzhugh–nagumo oscillators, International Journal of Electronics 101 (2) (2014) 220–230.

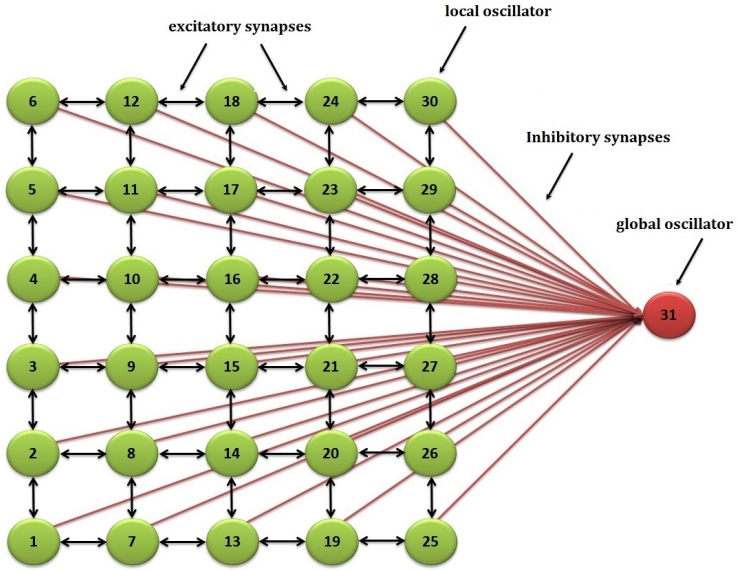


Figure 1: A 2-D LEGION network with four-neighborhood connections and a global inhibitor (only 31 neurons are shown to simplify the drawing).

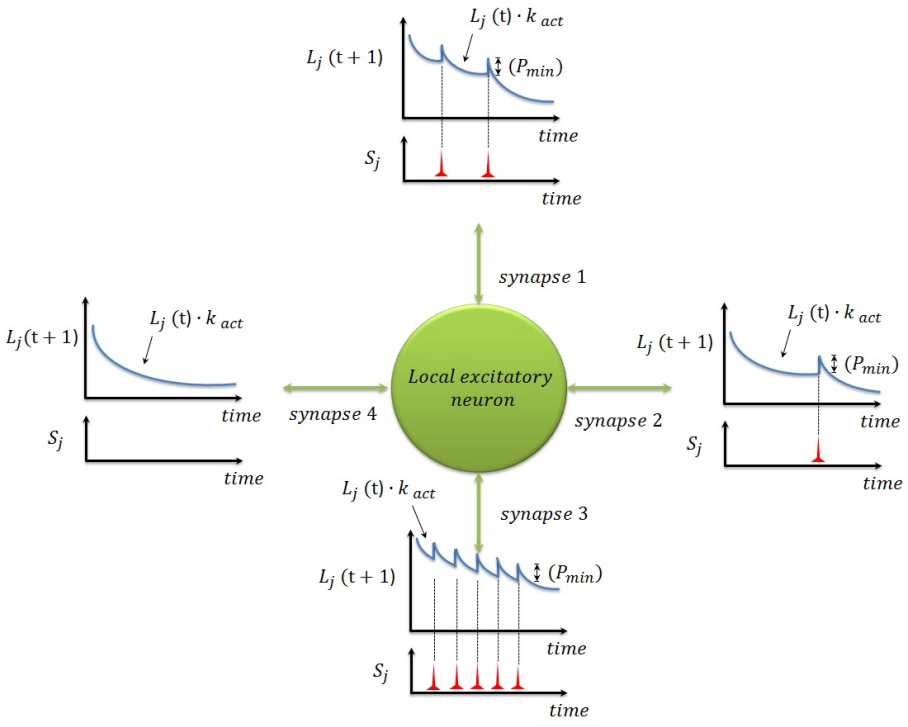


Figure 2: Local excitatory neuron with four excitatory synapses with the inclusion of synaptic variable $L_{ji}(t)$ to increase or decrease the synaptic potential $P(t)$.

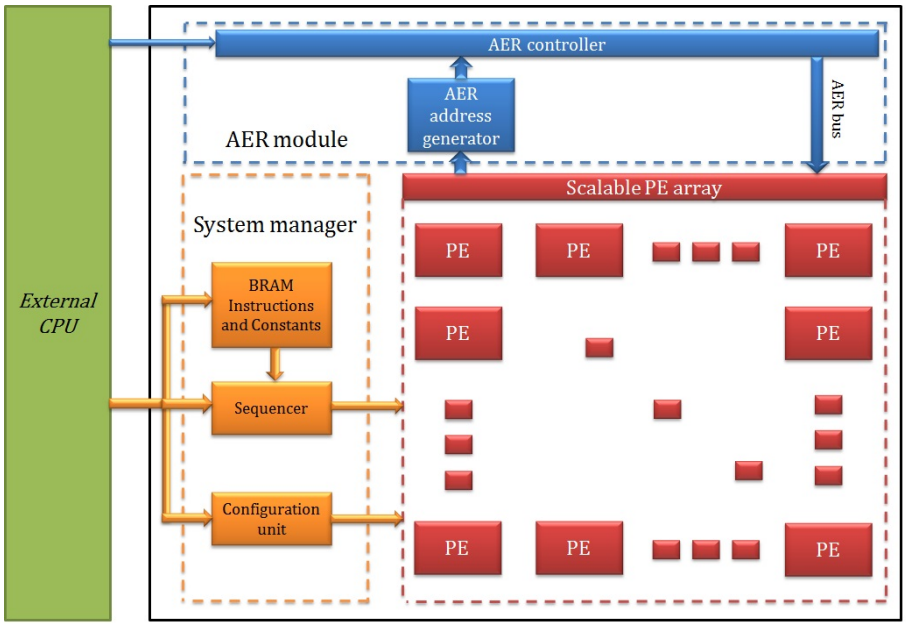


Figure 3: Architectural overview of neuromorphic architecture.

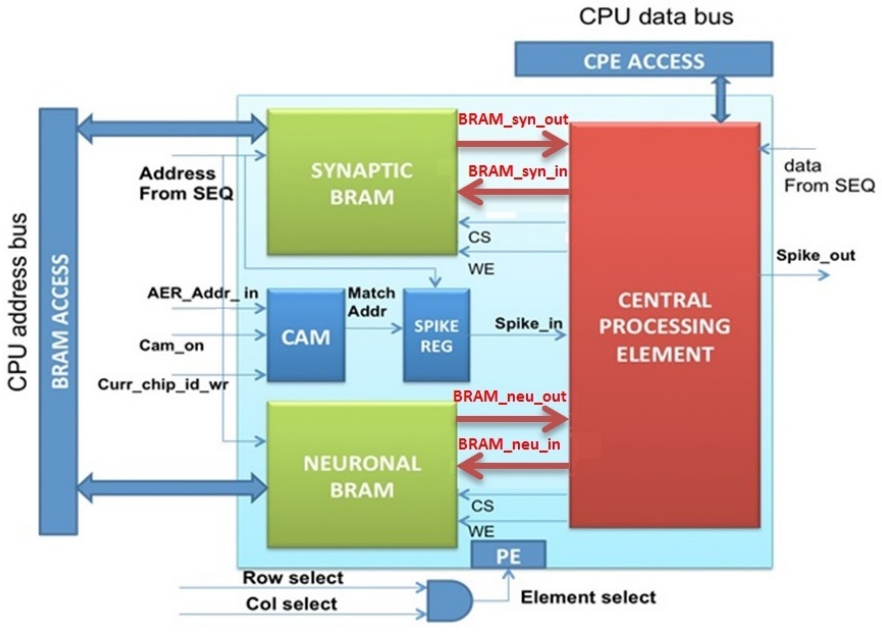


Figure 4: Structure of a Processing Element (PE).

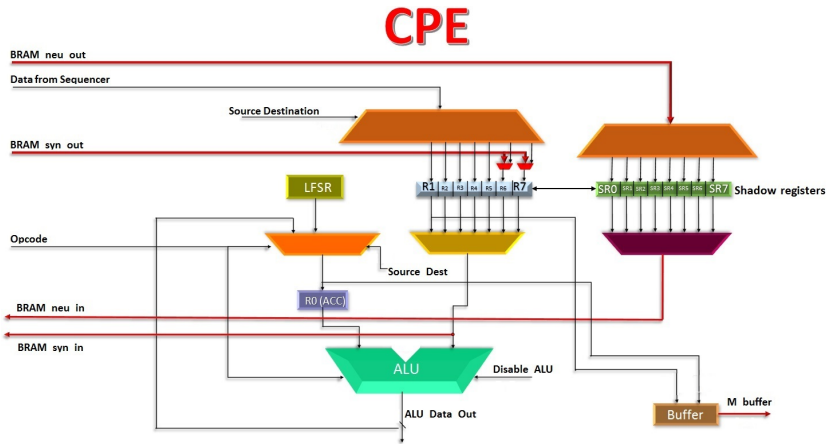


Figure 5: Central Processing Element data path

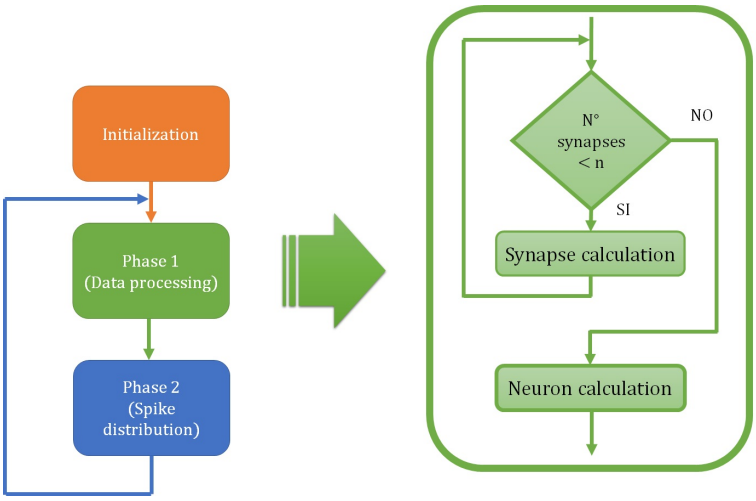


Figure 6: Execution loop for SNN emulation.

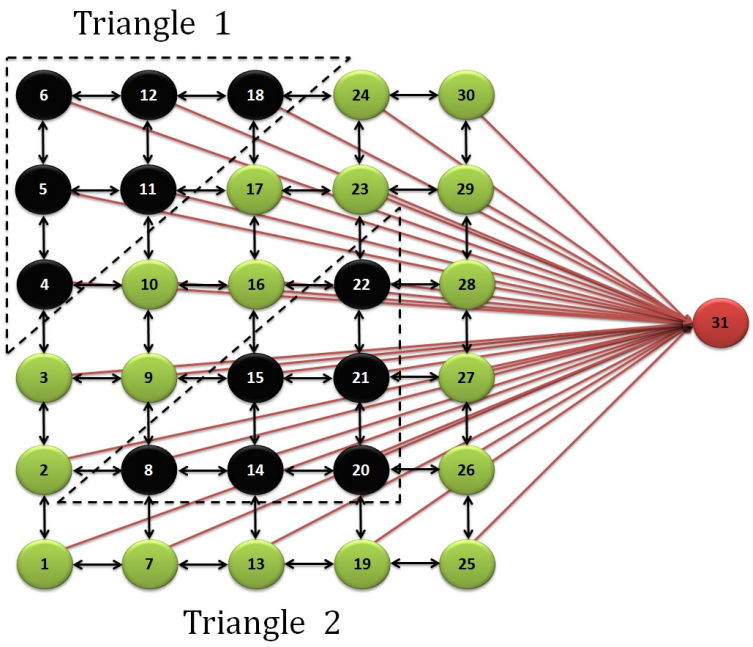


Figure 7: Image with two triangles

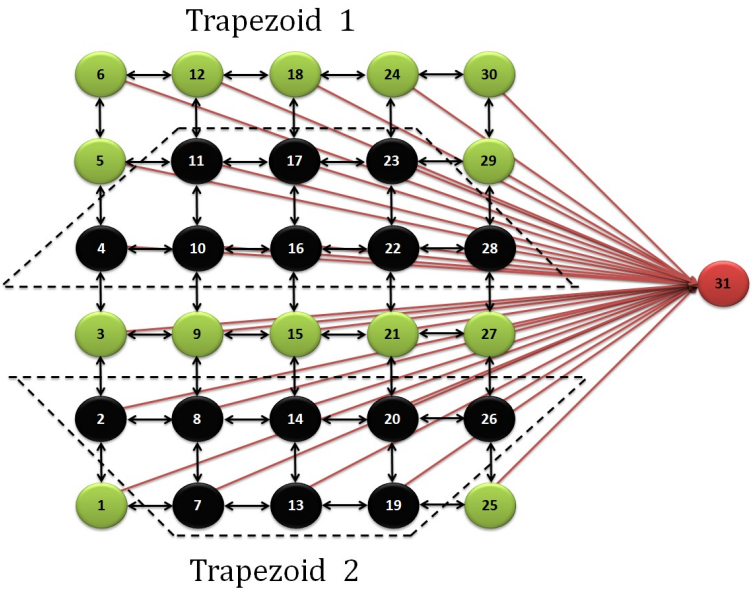


Figure 8: Image with two trapezoids

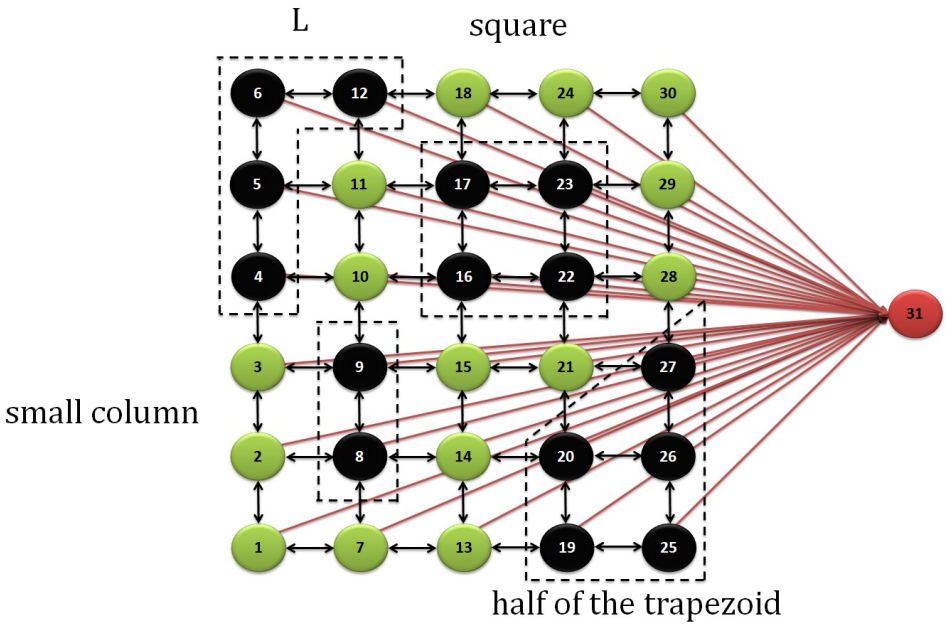


Figure 9: Image with four figures (letter L, square, small column and the half of the trapezoid)

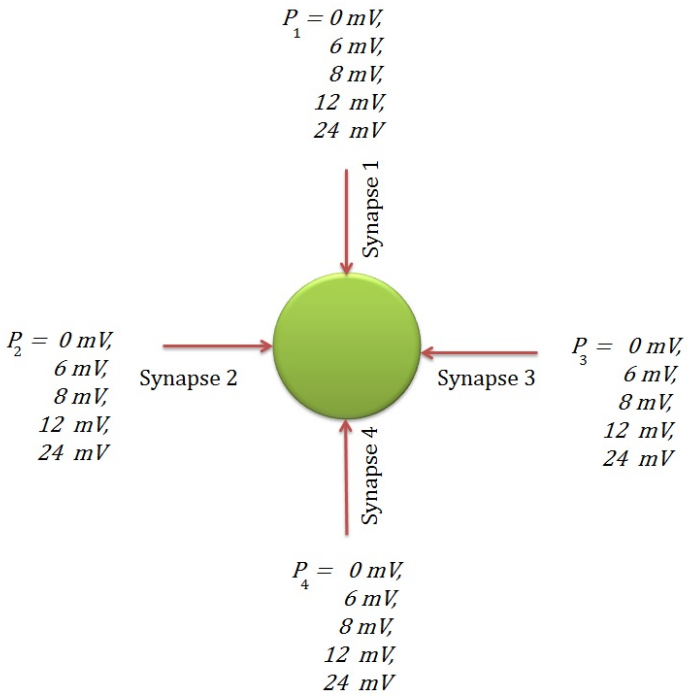


Figure 10: Synaptic potential distribution for the normalization process of the synaptic weights

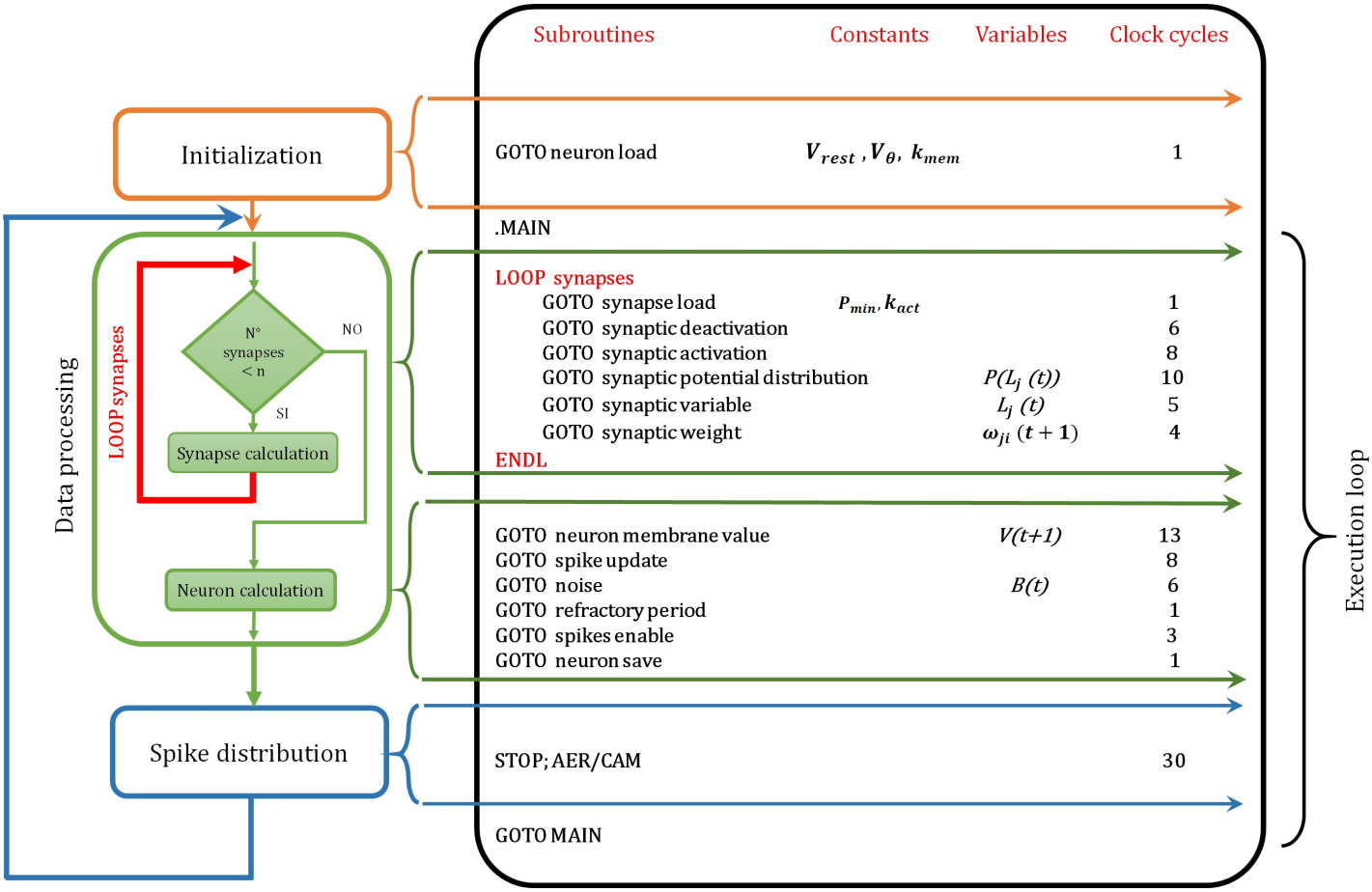


Figure 11: Main program in assembly code for the SNN-LEGION simulation

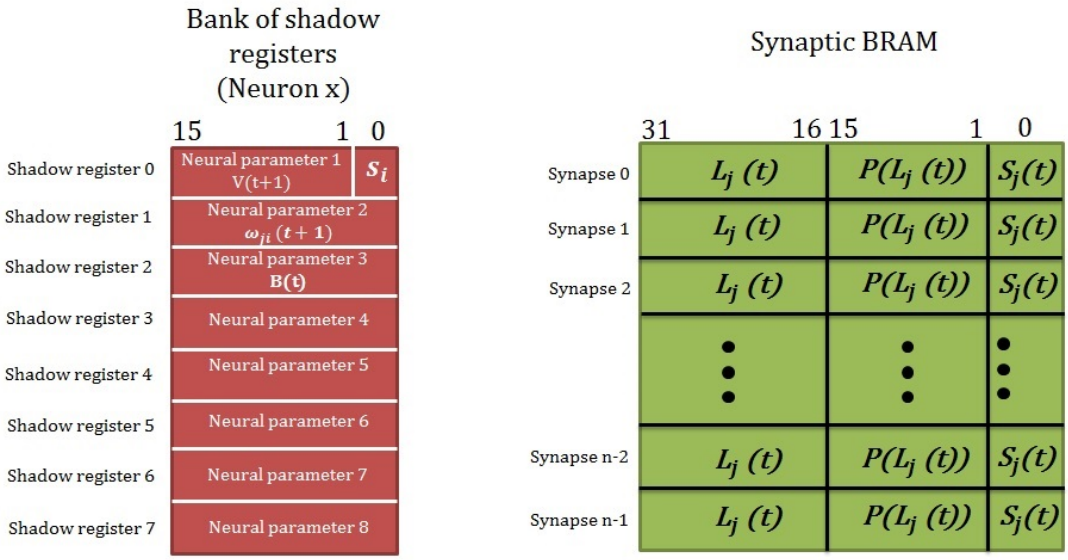


Figure 12: The allocation of neural/synaptic parameters on the optimized neuromorphic architecture.

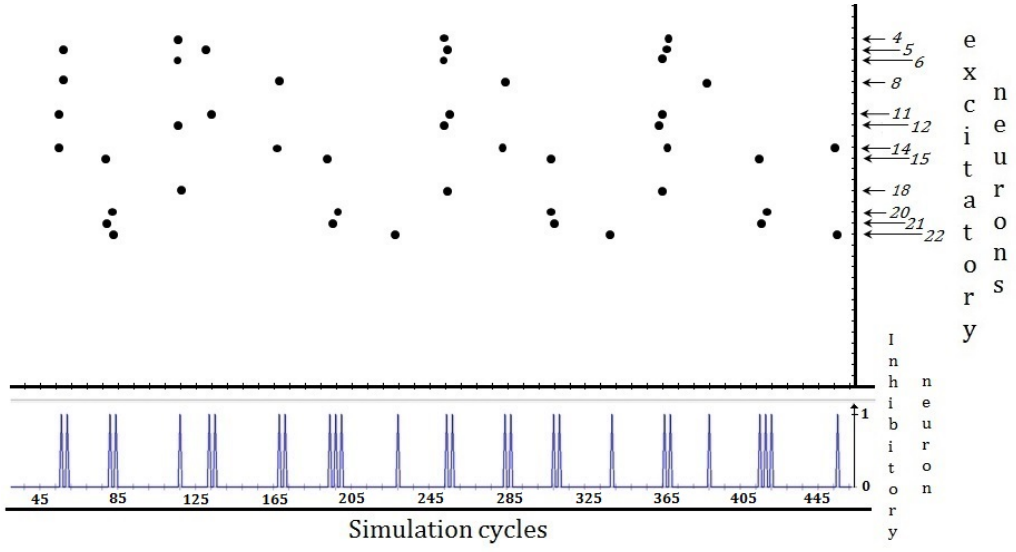


Figure 13: Initial neural activity.

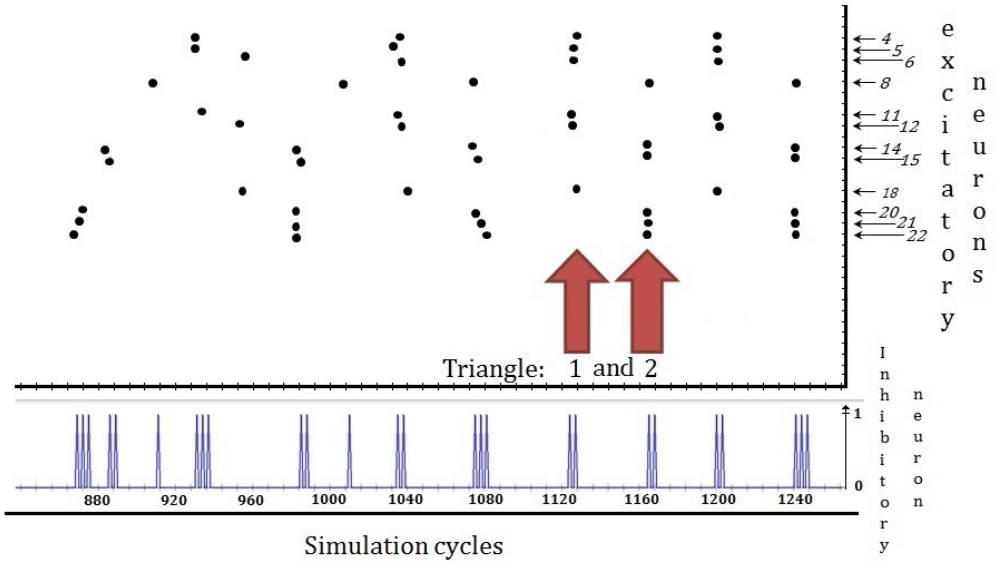


Figure 14: Neural activity showing the segmentation of the triangles.

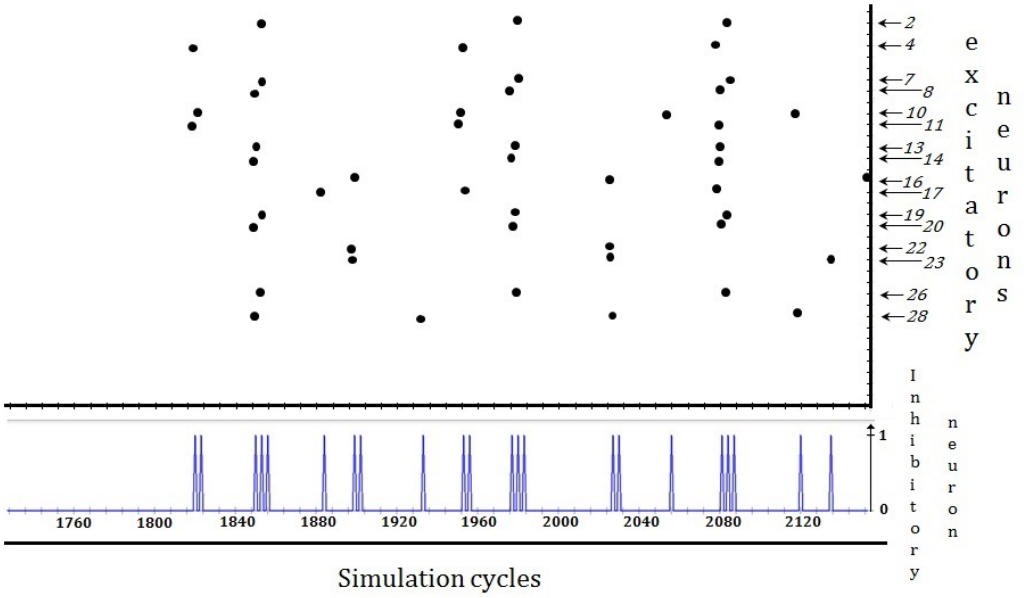


Figure 15: Neural activity showing the self-adapt of the SNN network to carry out the segmentation of the trapezoids.

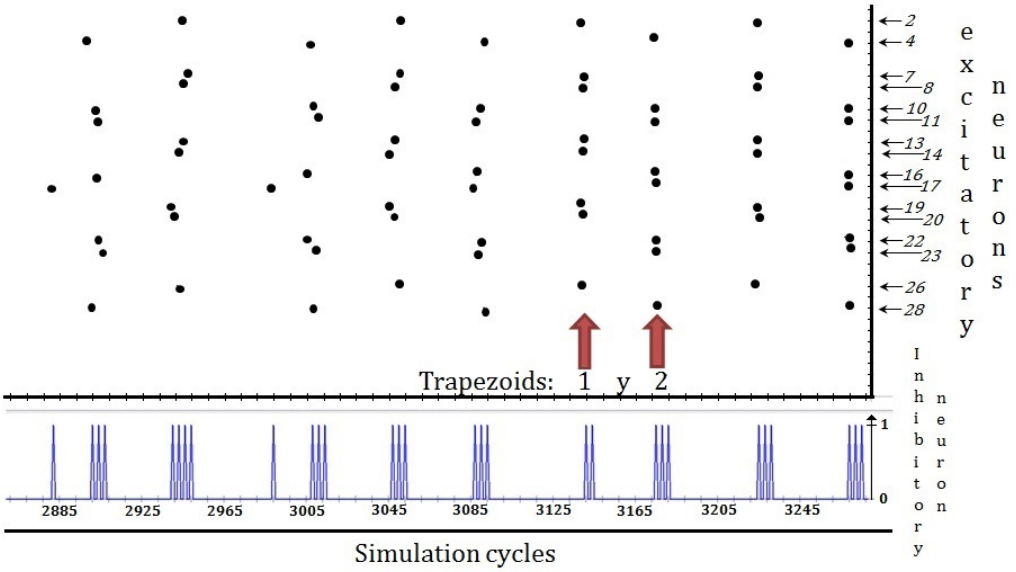


Figure 16: Neural activity showing the segmentation of the trapezoids.

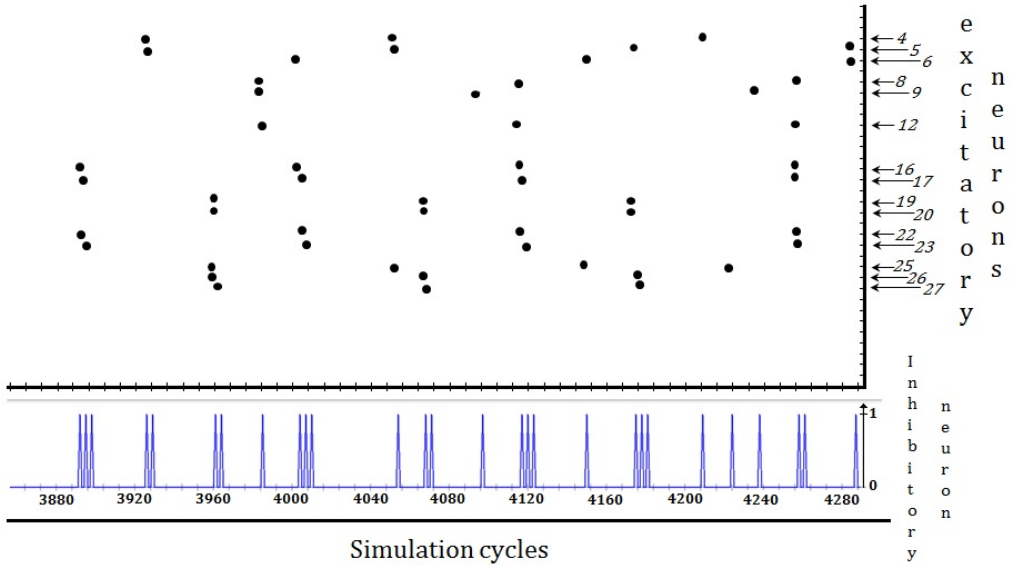


Figure 17: Neural activity showing the self-adapt the SNN network to carry out the segmentation of a set of objects.

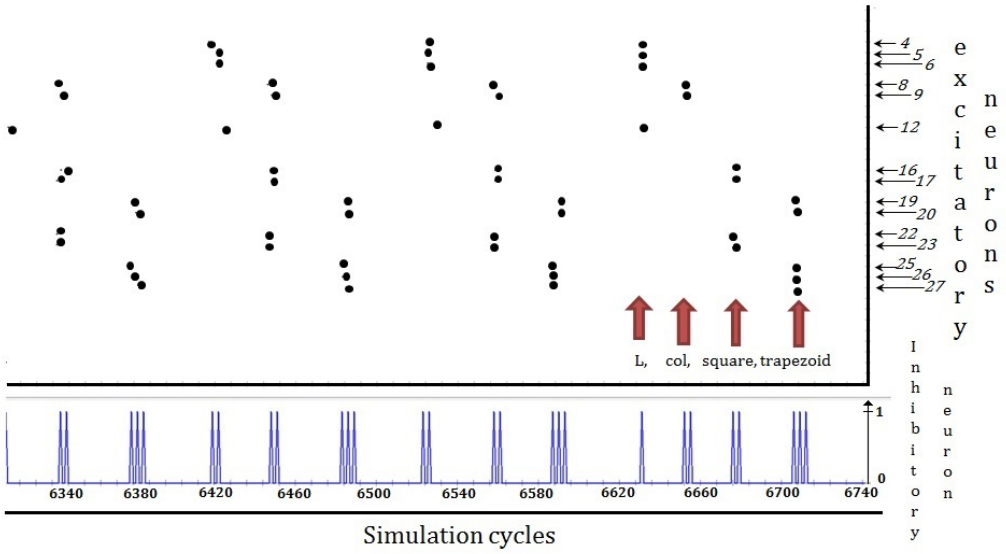
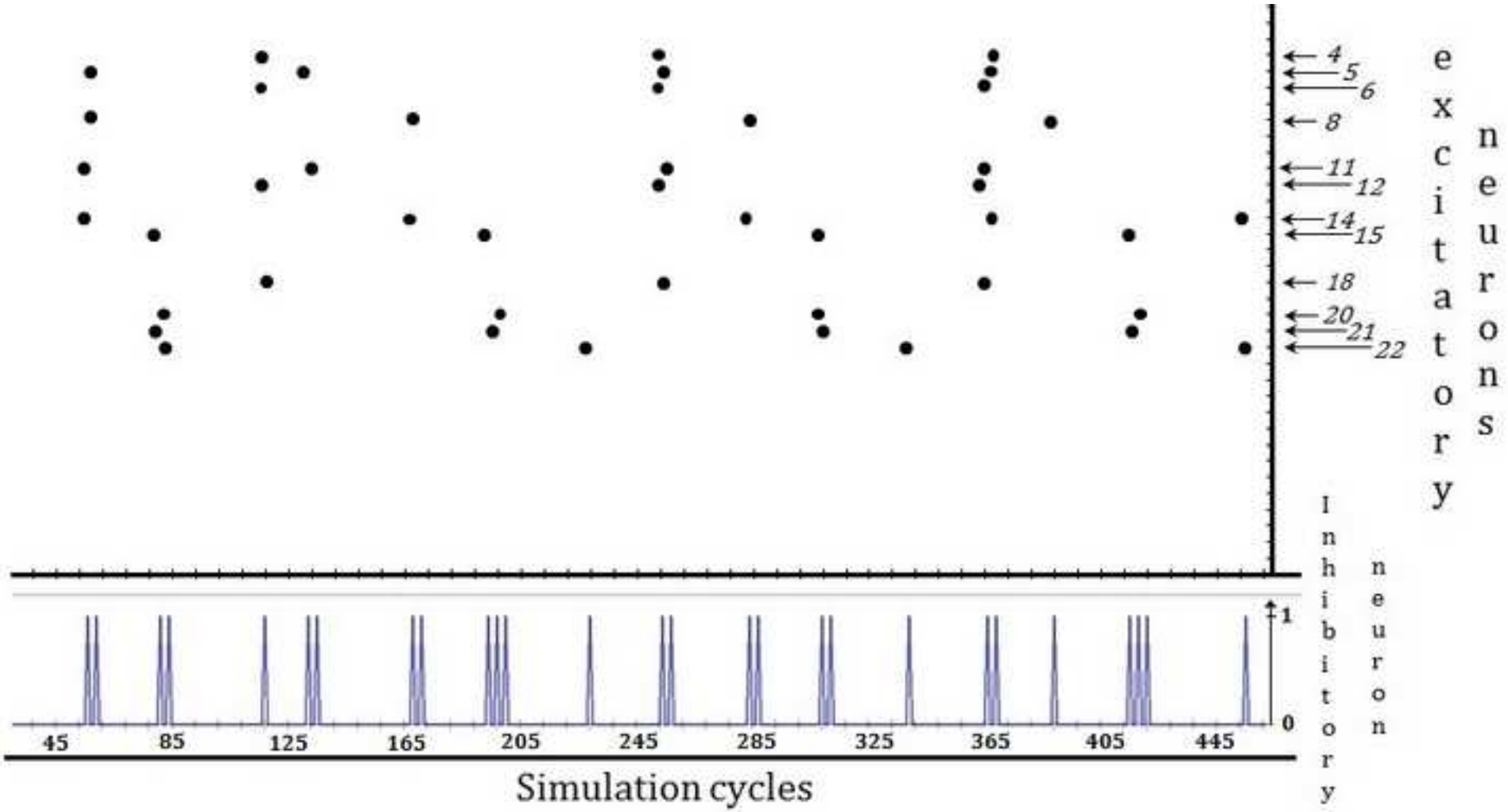


Figure 18: Neural activity showing the segmentation of a set of objects.

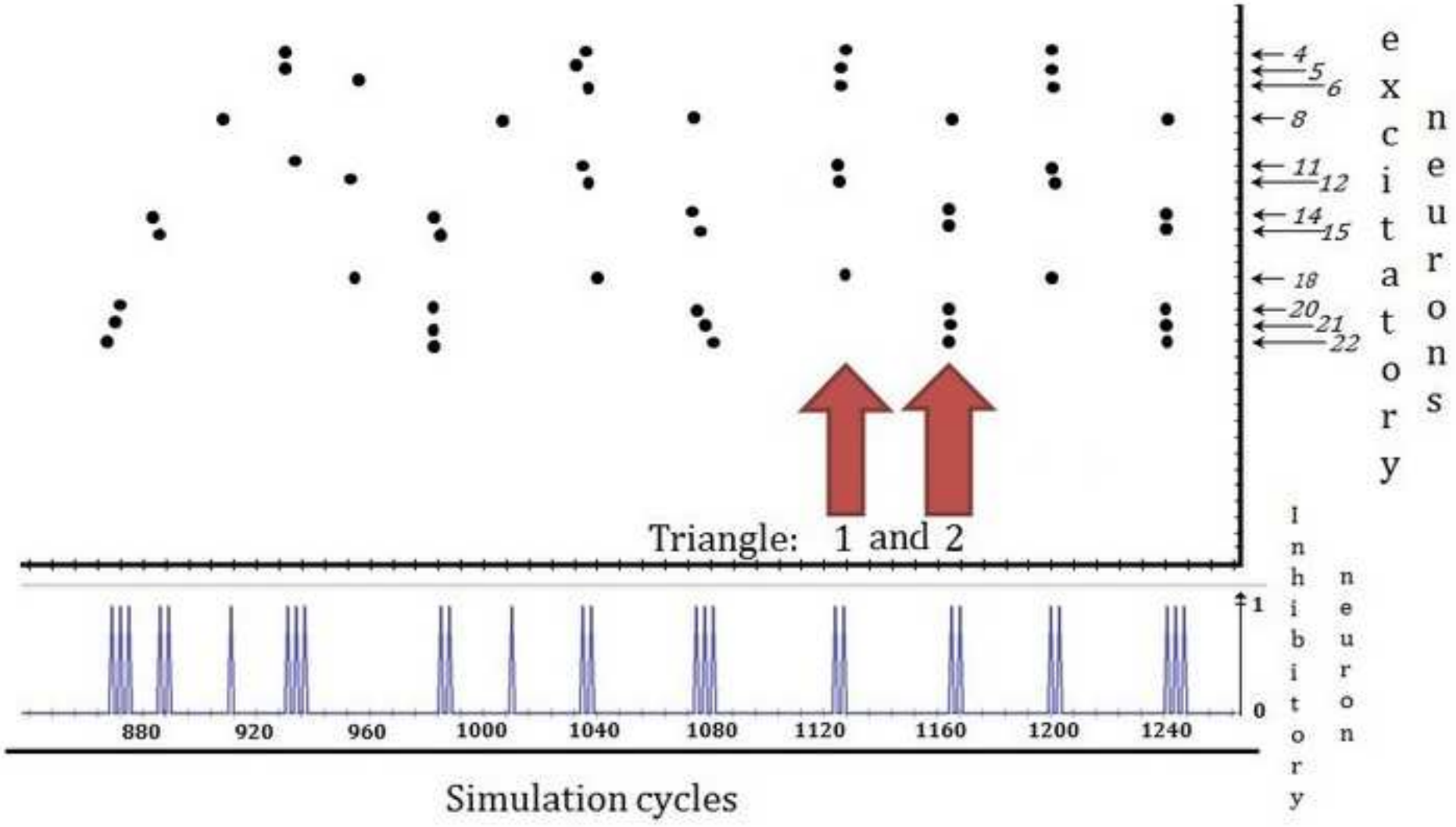
Figure

Click here to download high resolution image



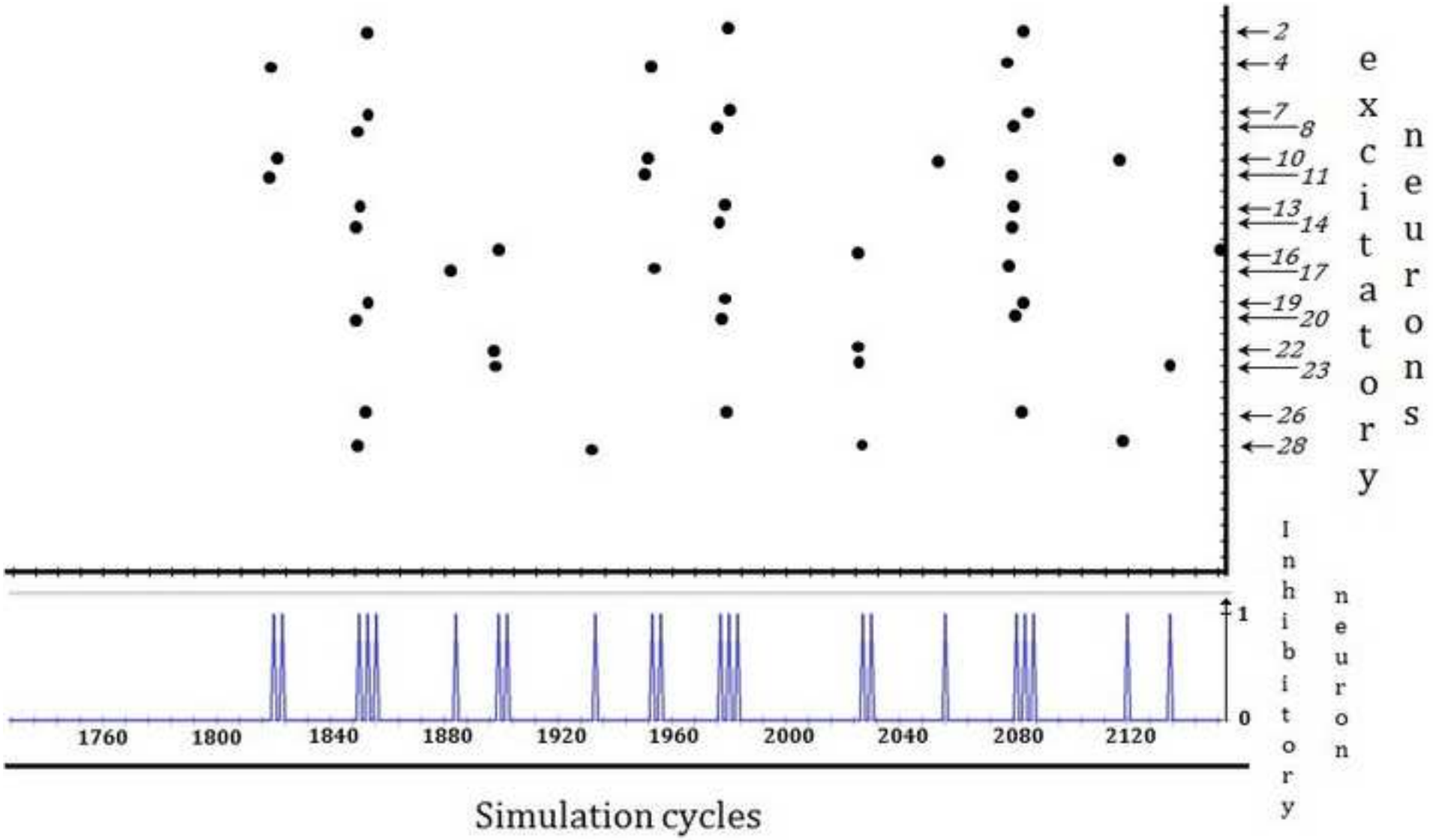
Figure

Click here to download high resolution image



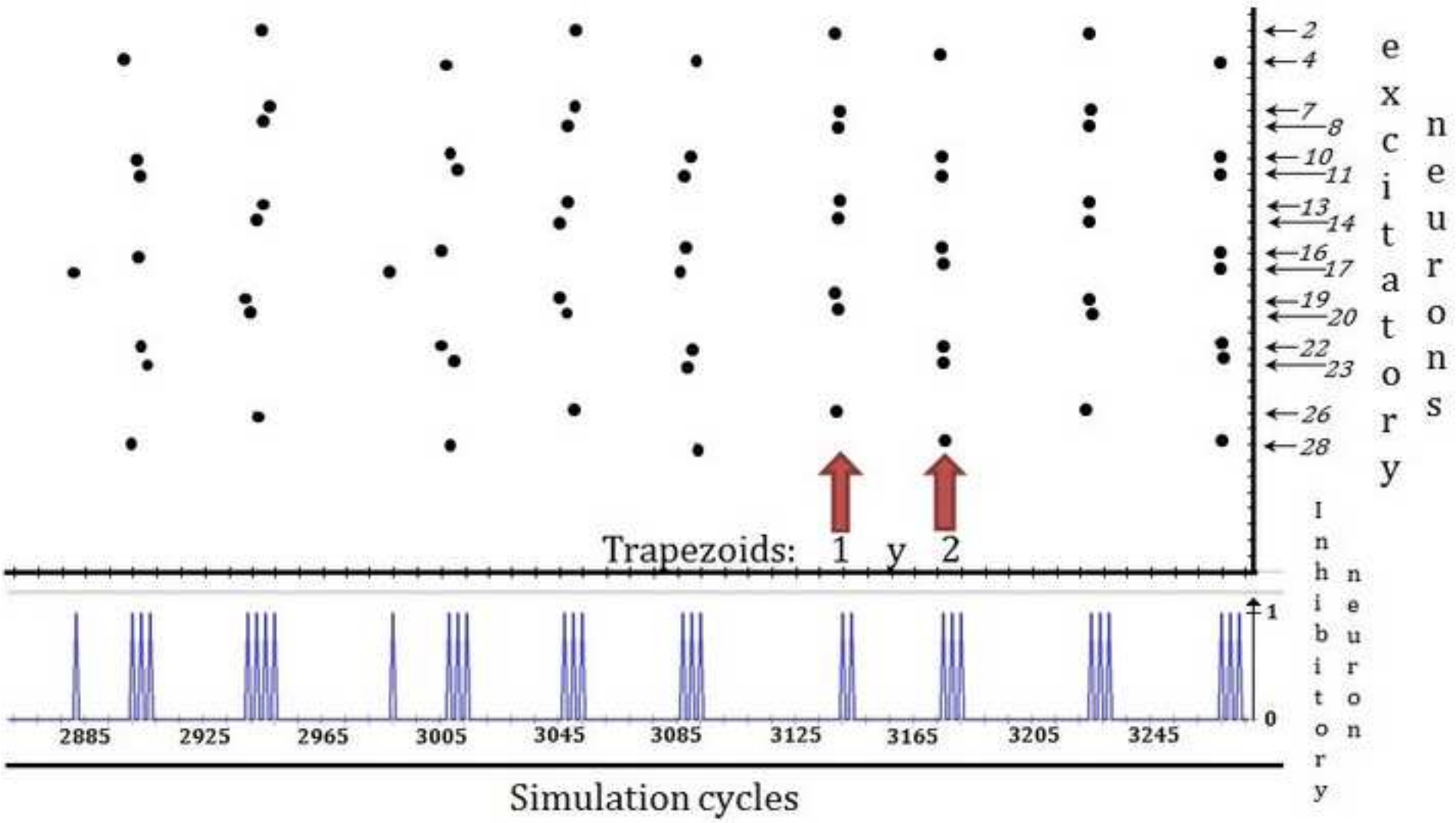
Figure

Click here to download high resolution image



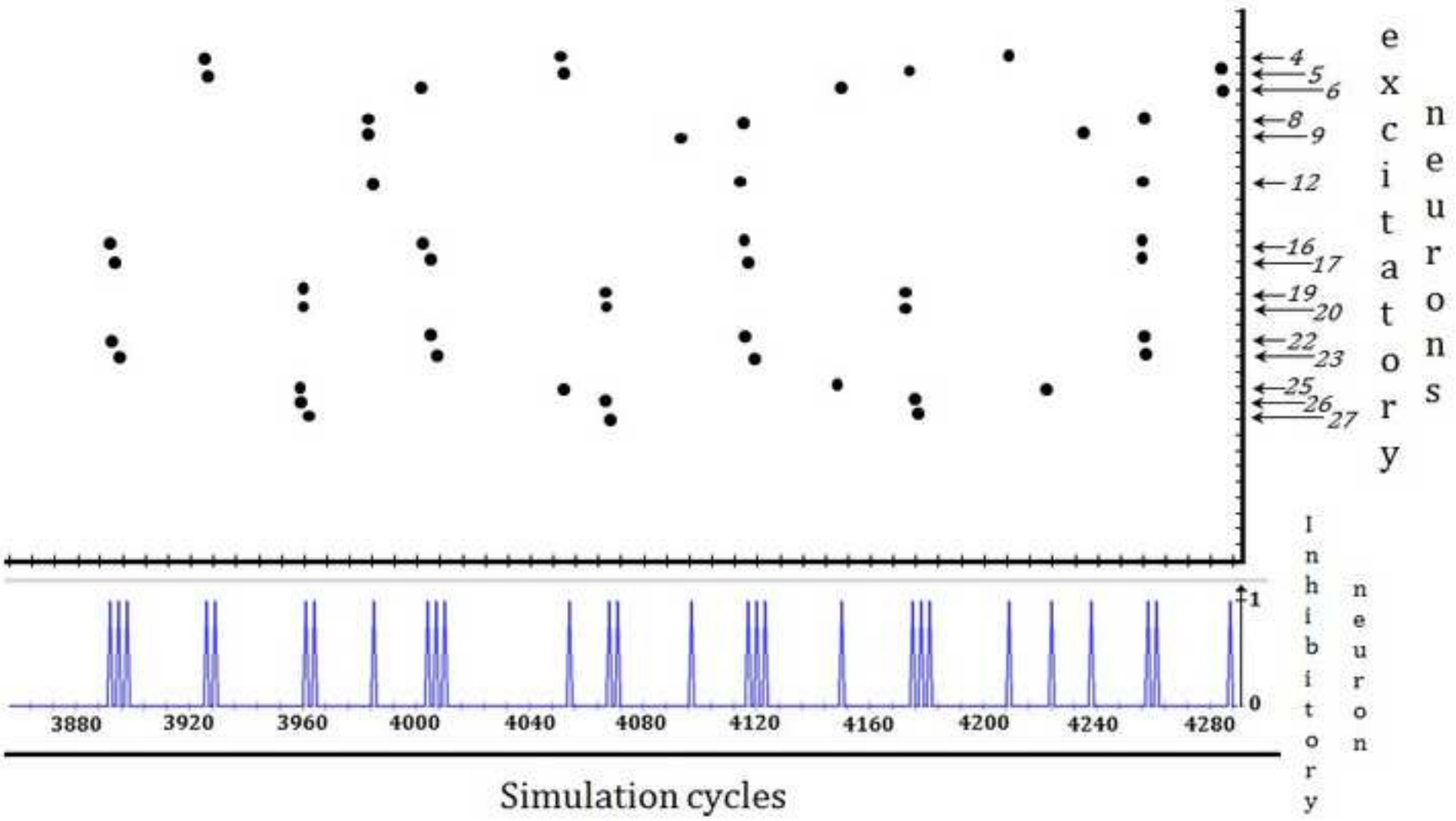
Figure

Click here to download high resolution image



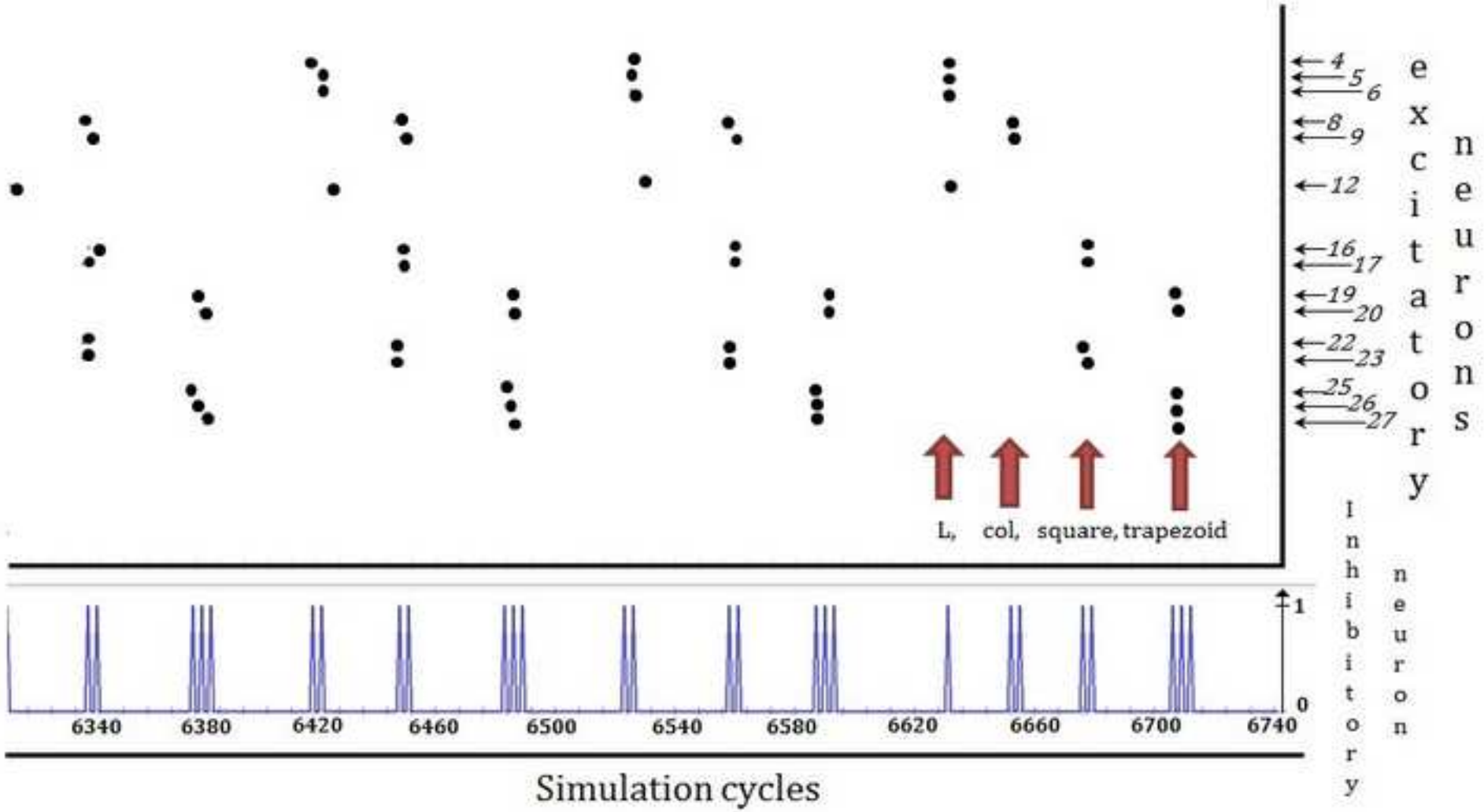
Figure

Click here to download high resolution image



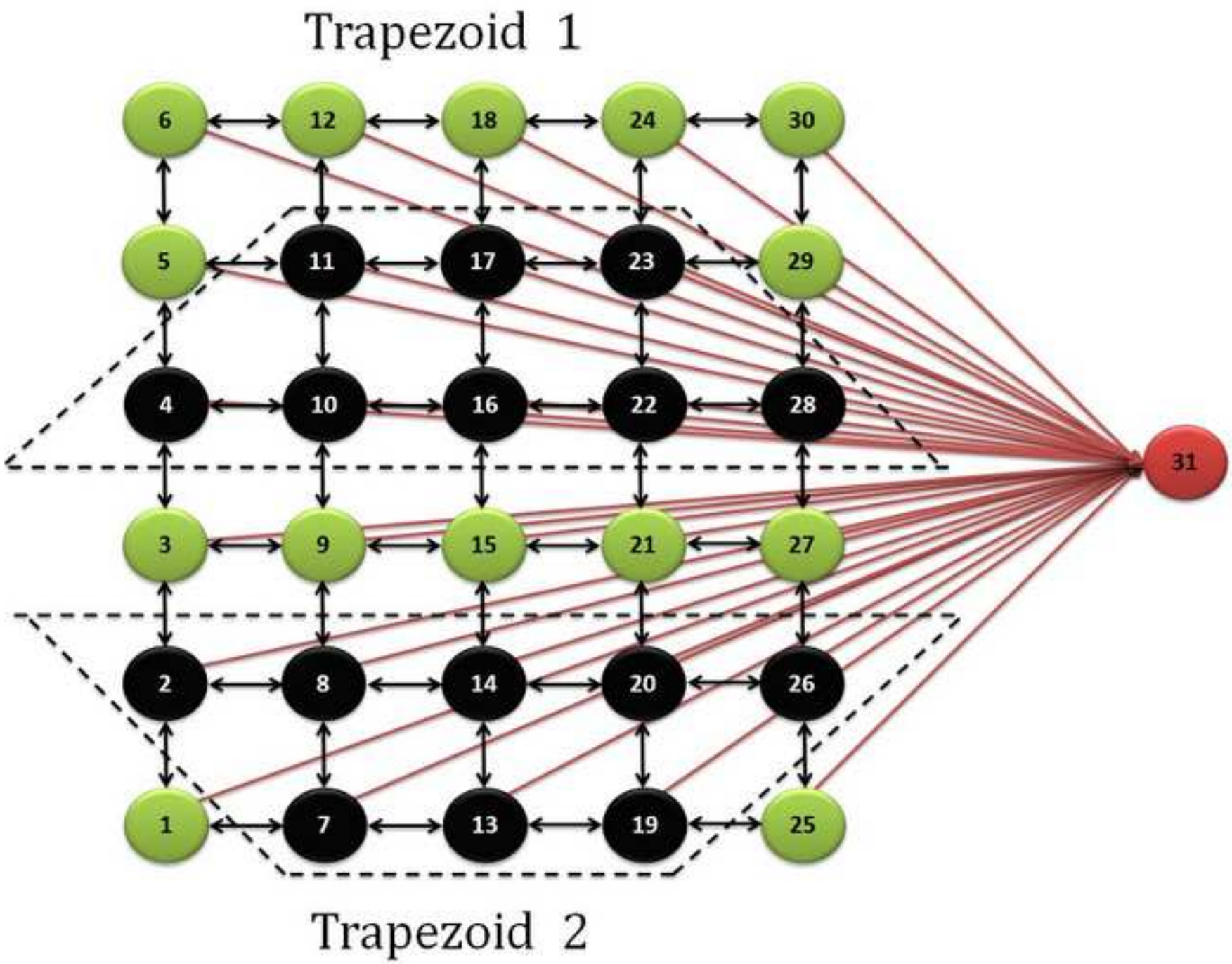
Figure

Click here to download high resolution image



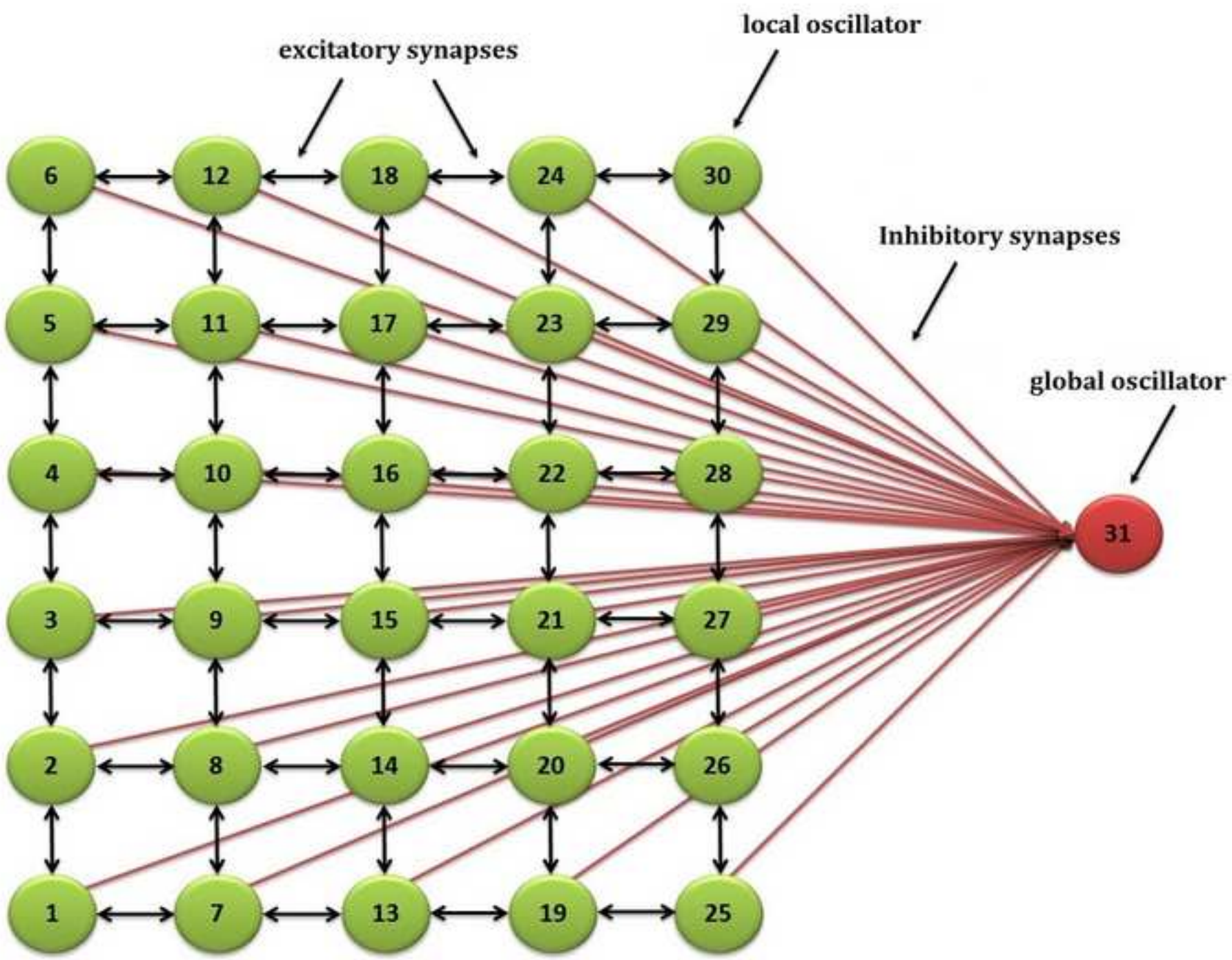
Figure

Click here to download high resolution image



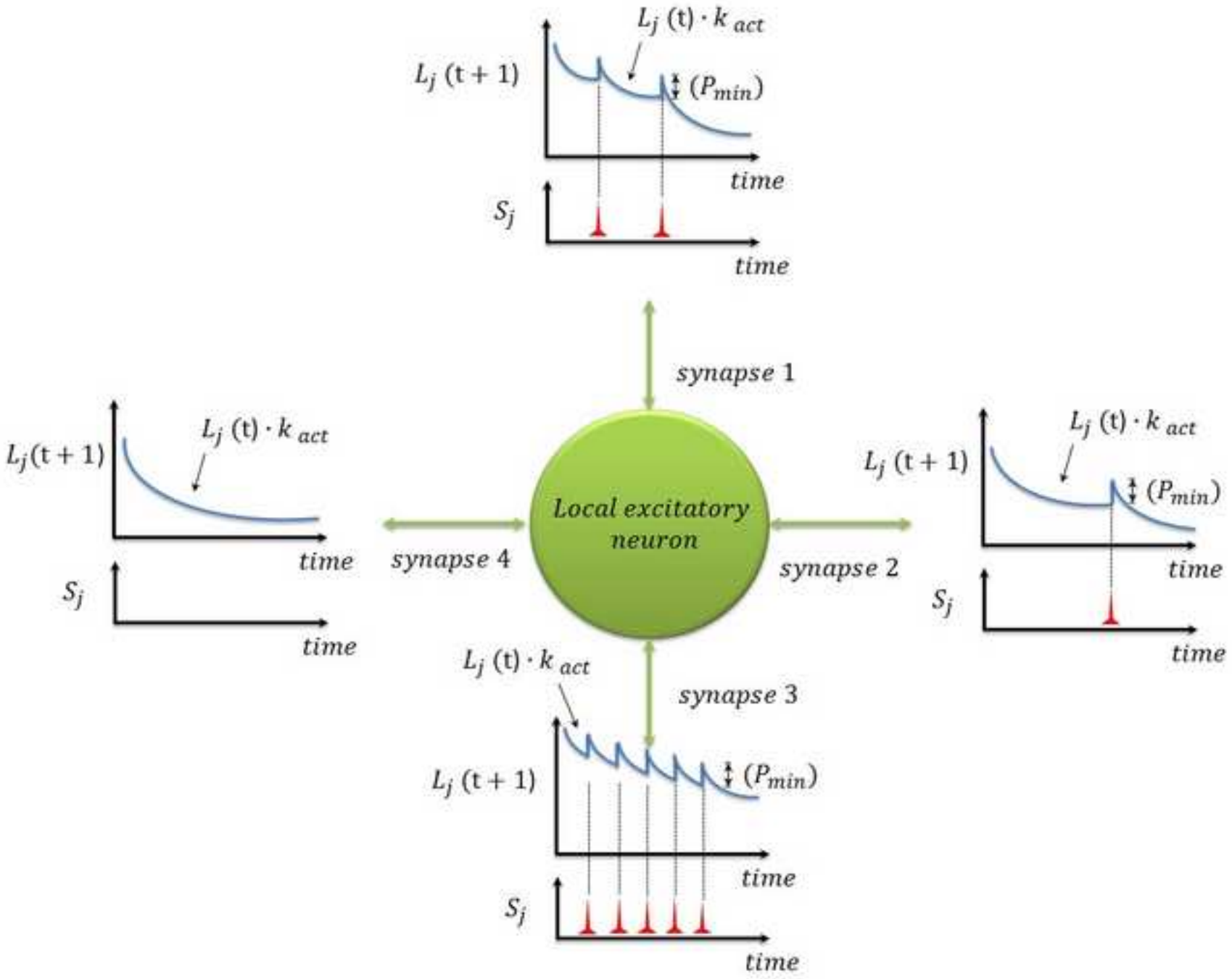
Figure

Click here to download high resolution image



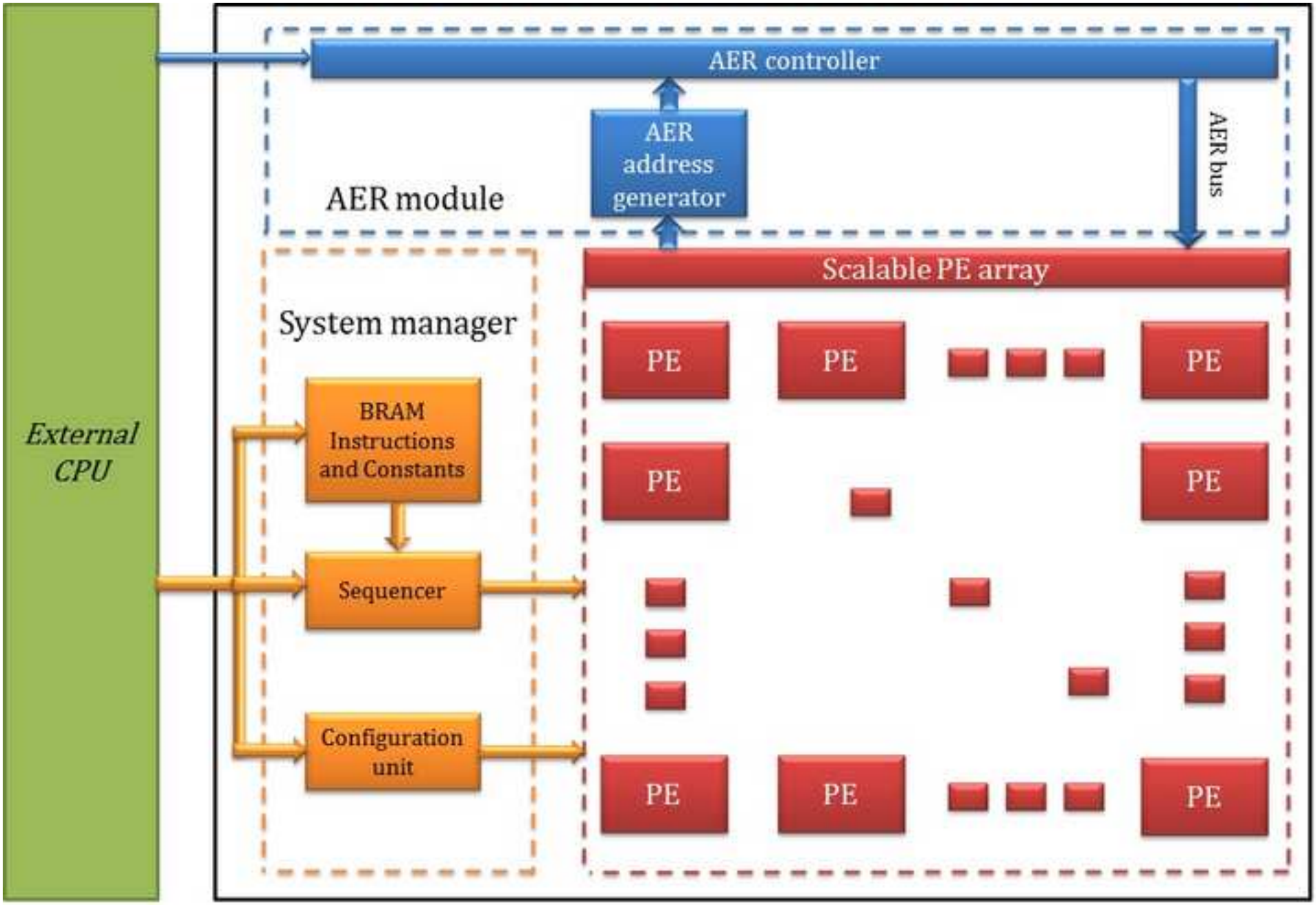
Figure

Click here to download high resolution image



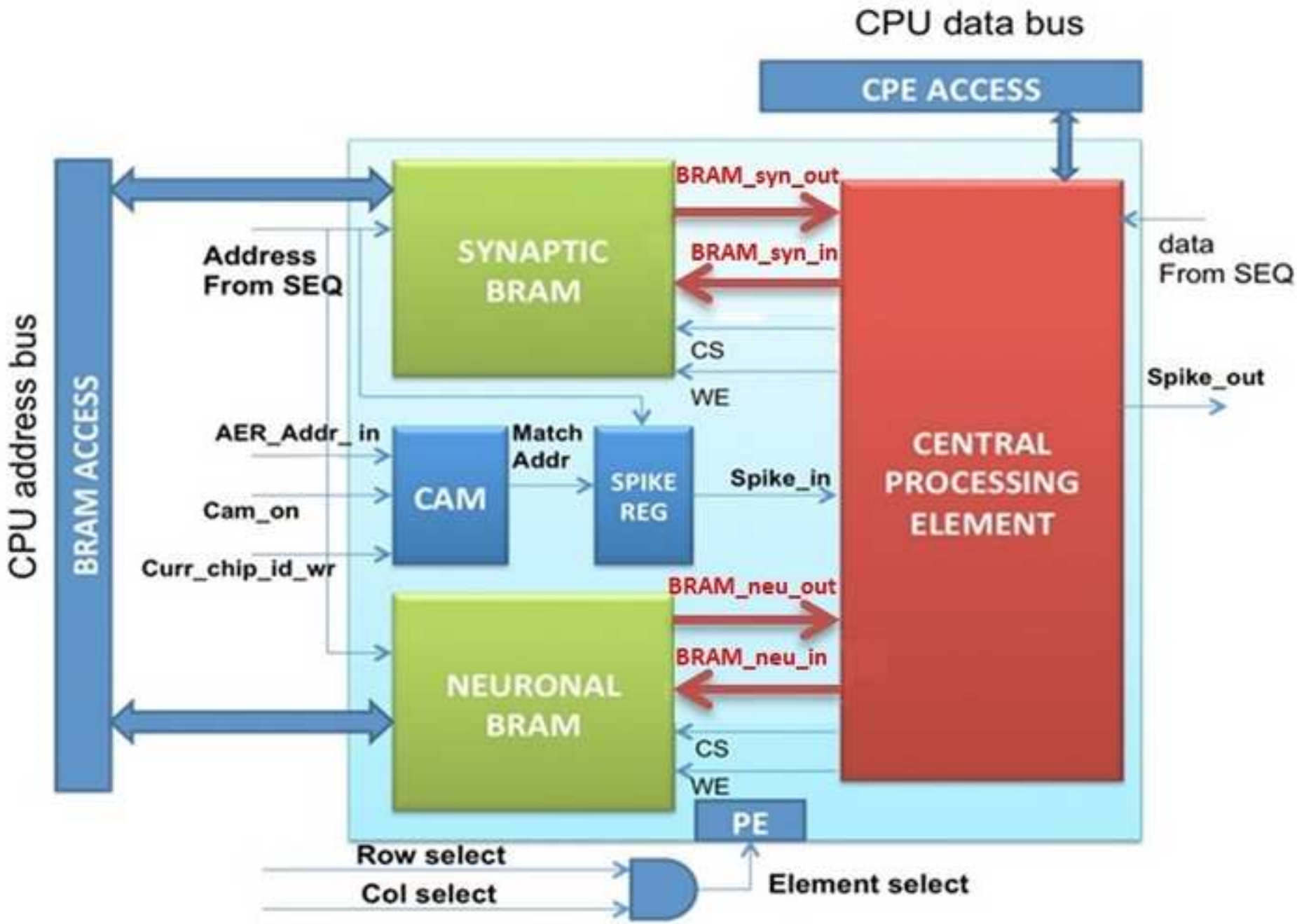
Figure

Click here to download high resolution image



Figure

Click here to download high resolution image



Figure

Click here to download high resolution image

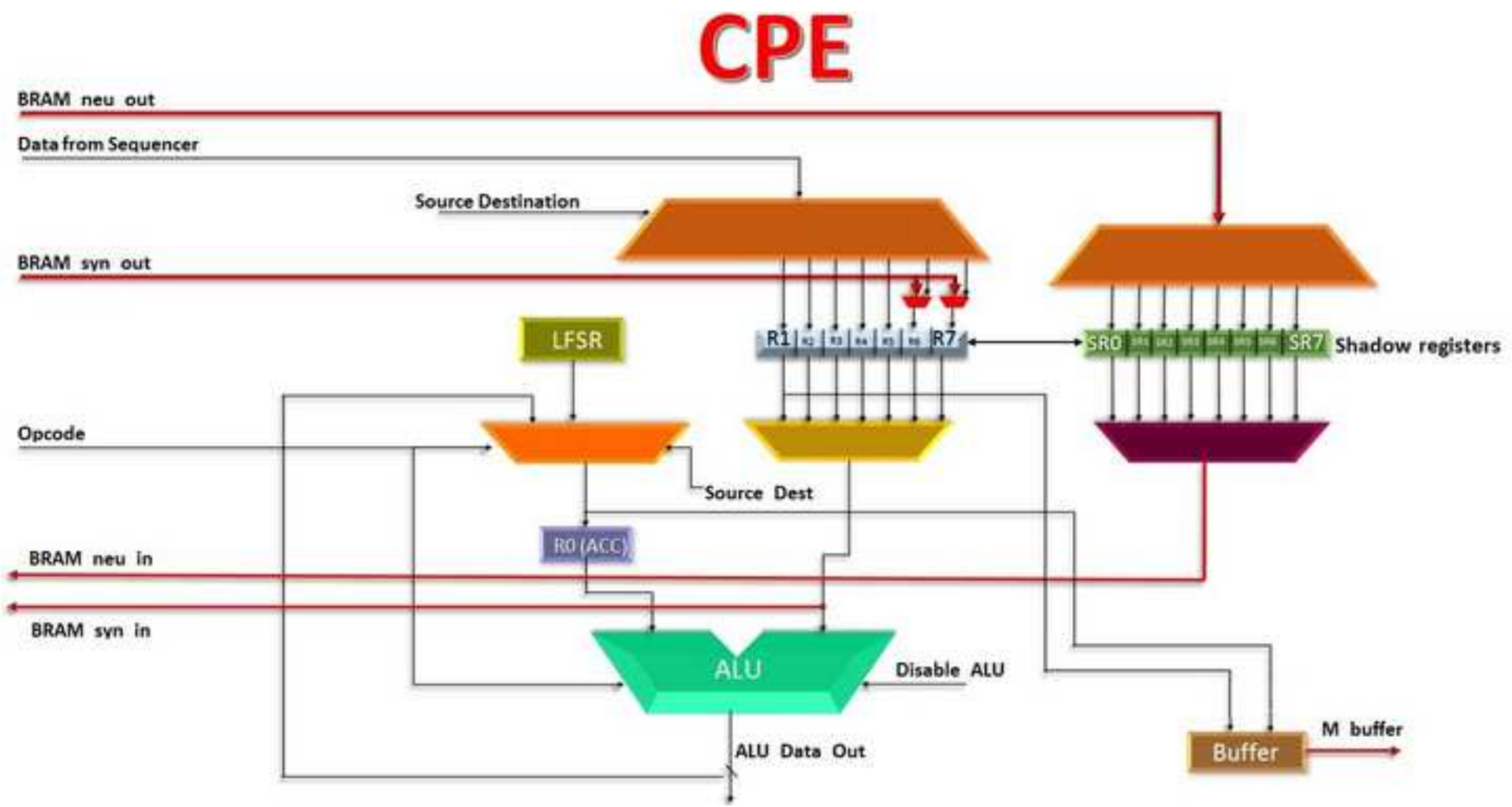
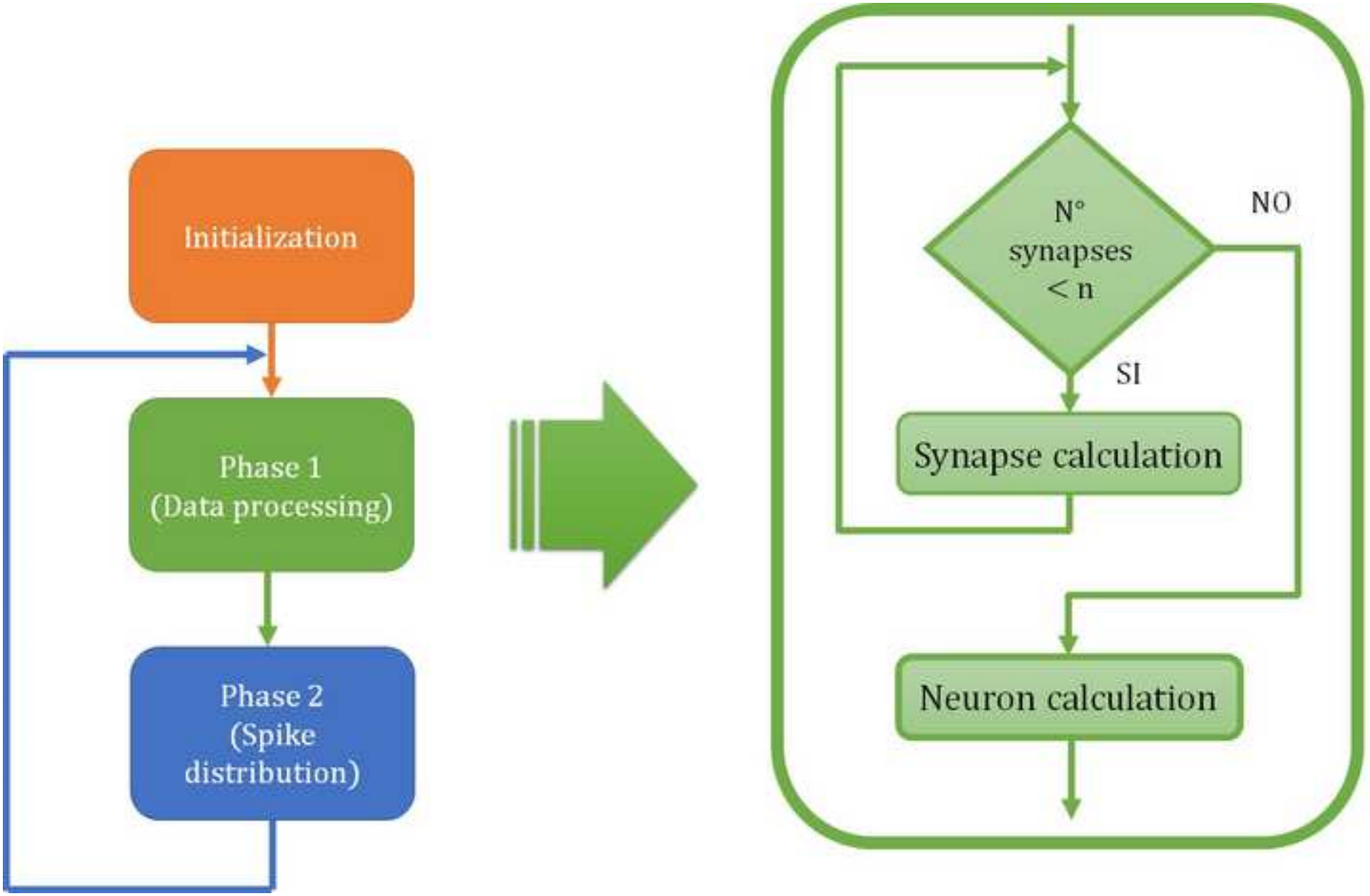
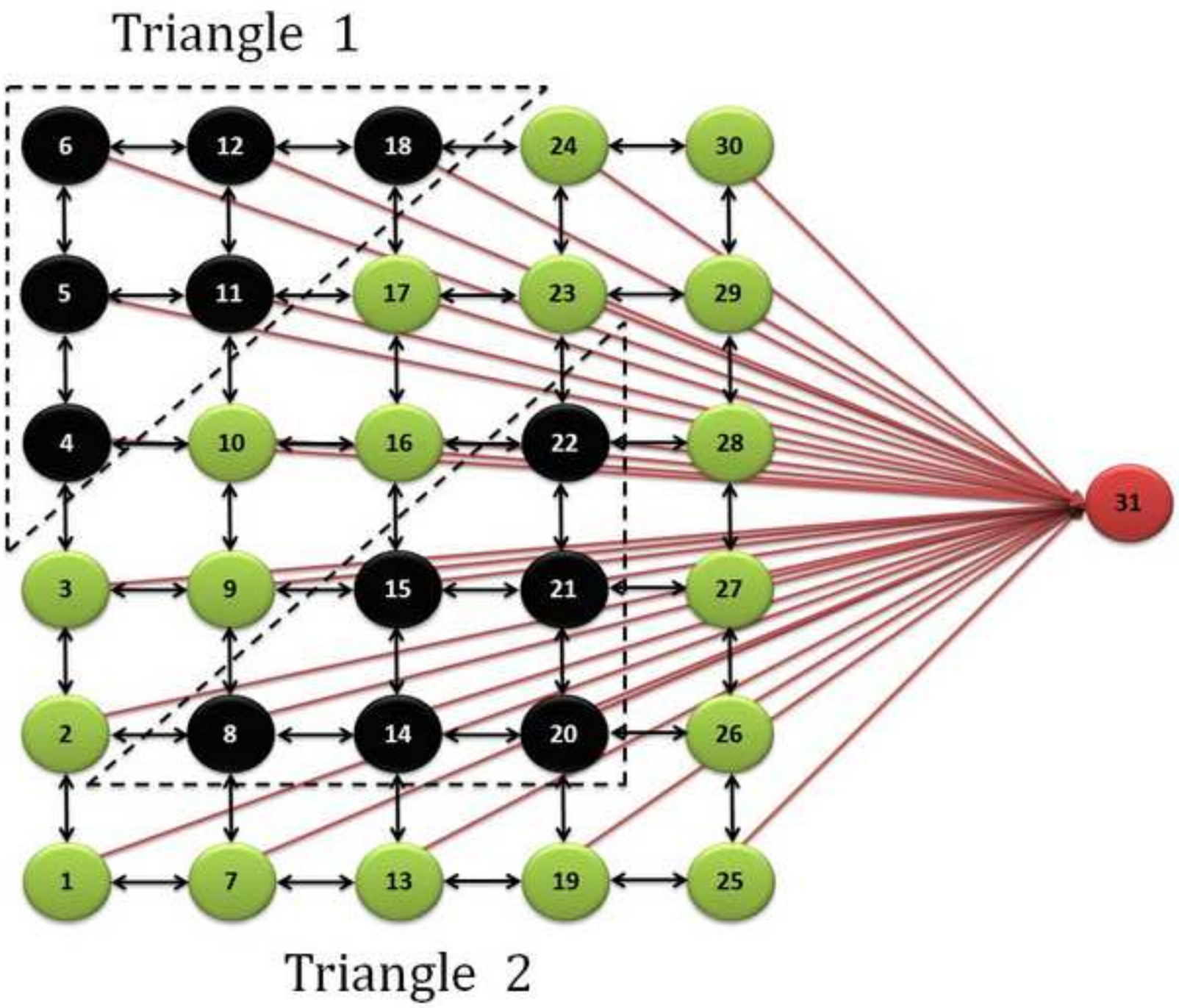


Figure
Click here to download high resolution image



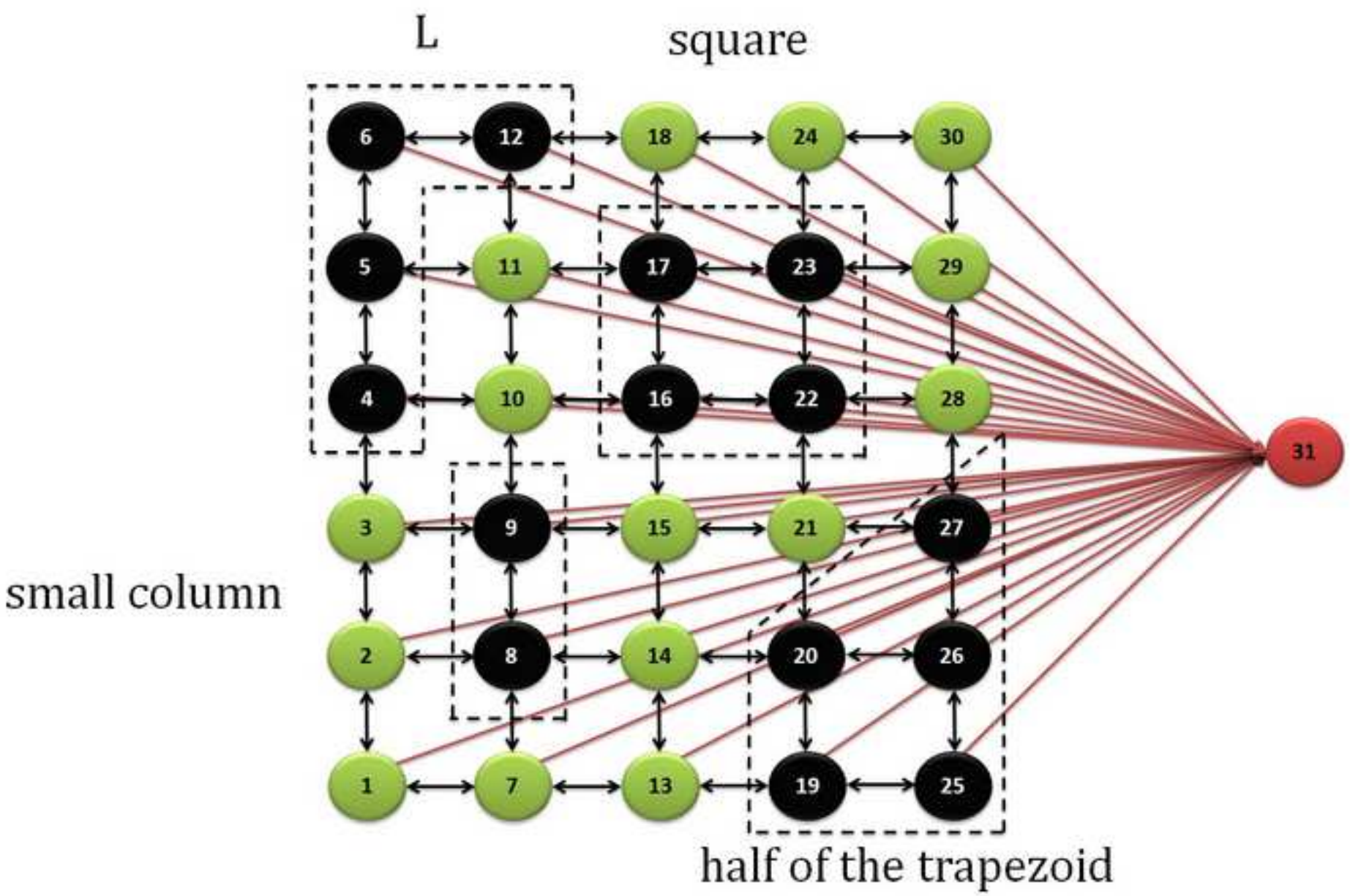
Figure

Click here to download high resolution image



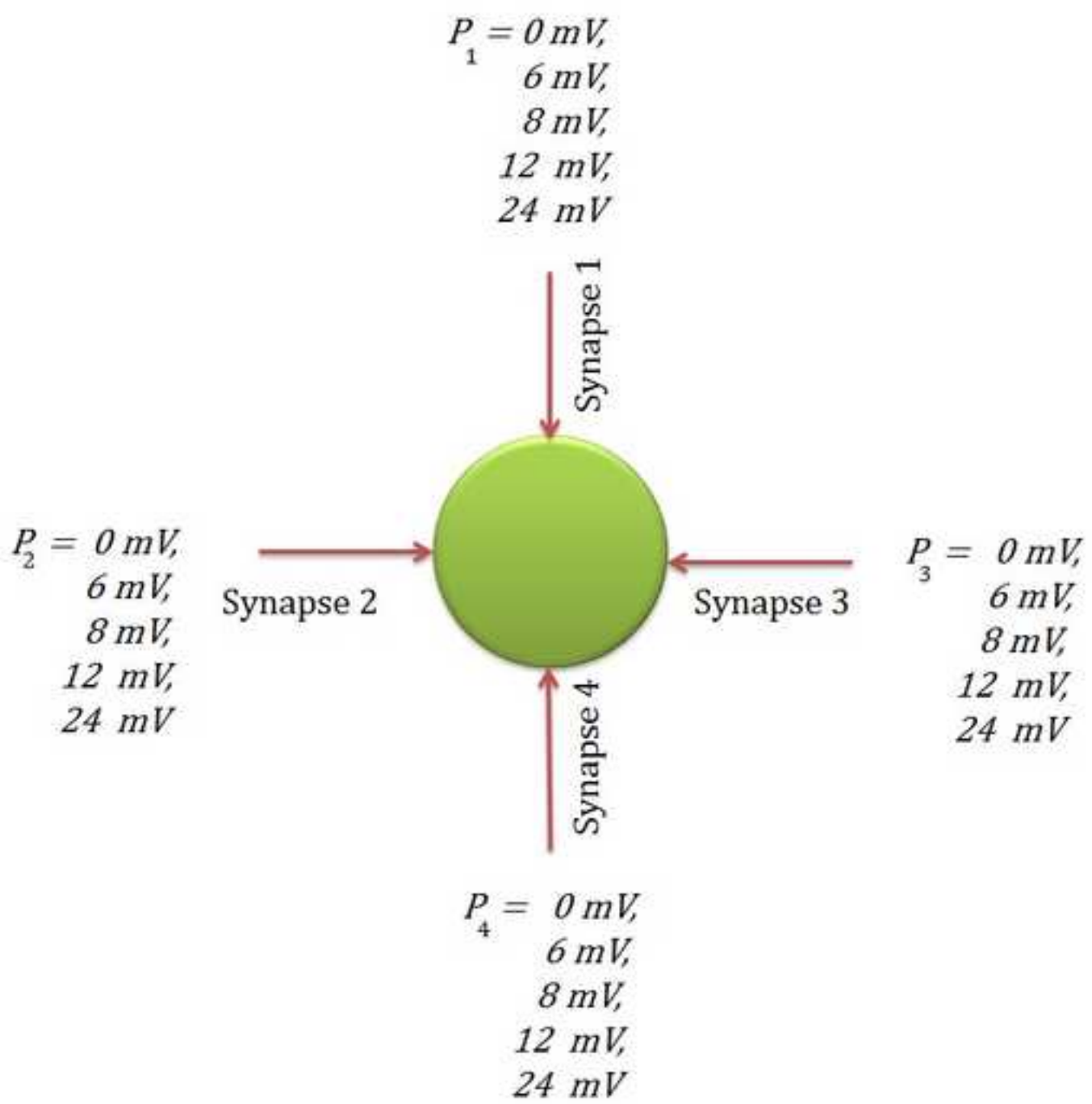
Figure

Click here to download high resolution image



Figure

Click here to download high resolution image



Figure

Click here to download high resolution image

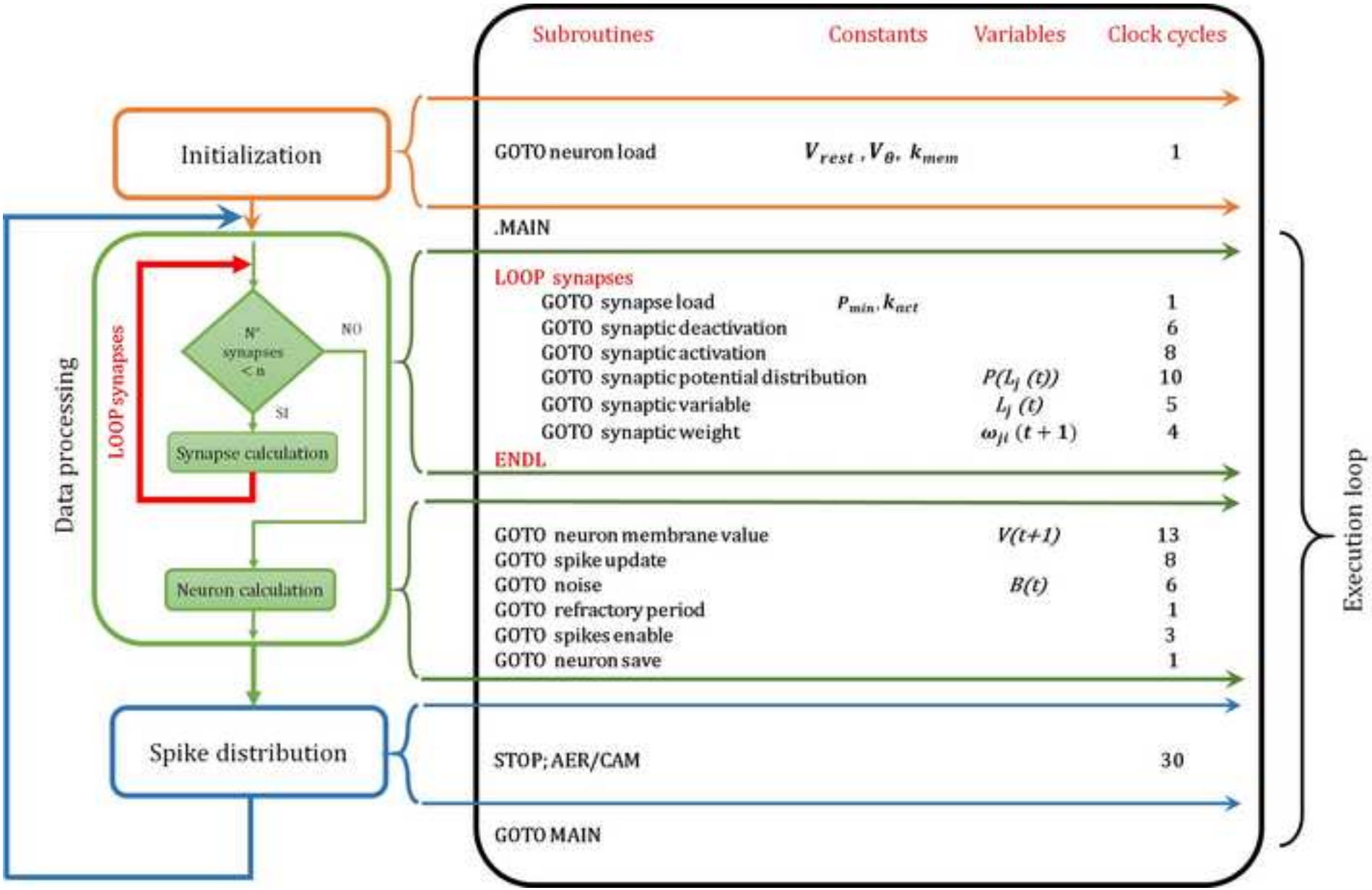
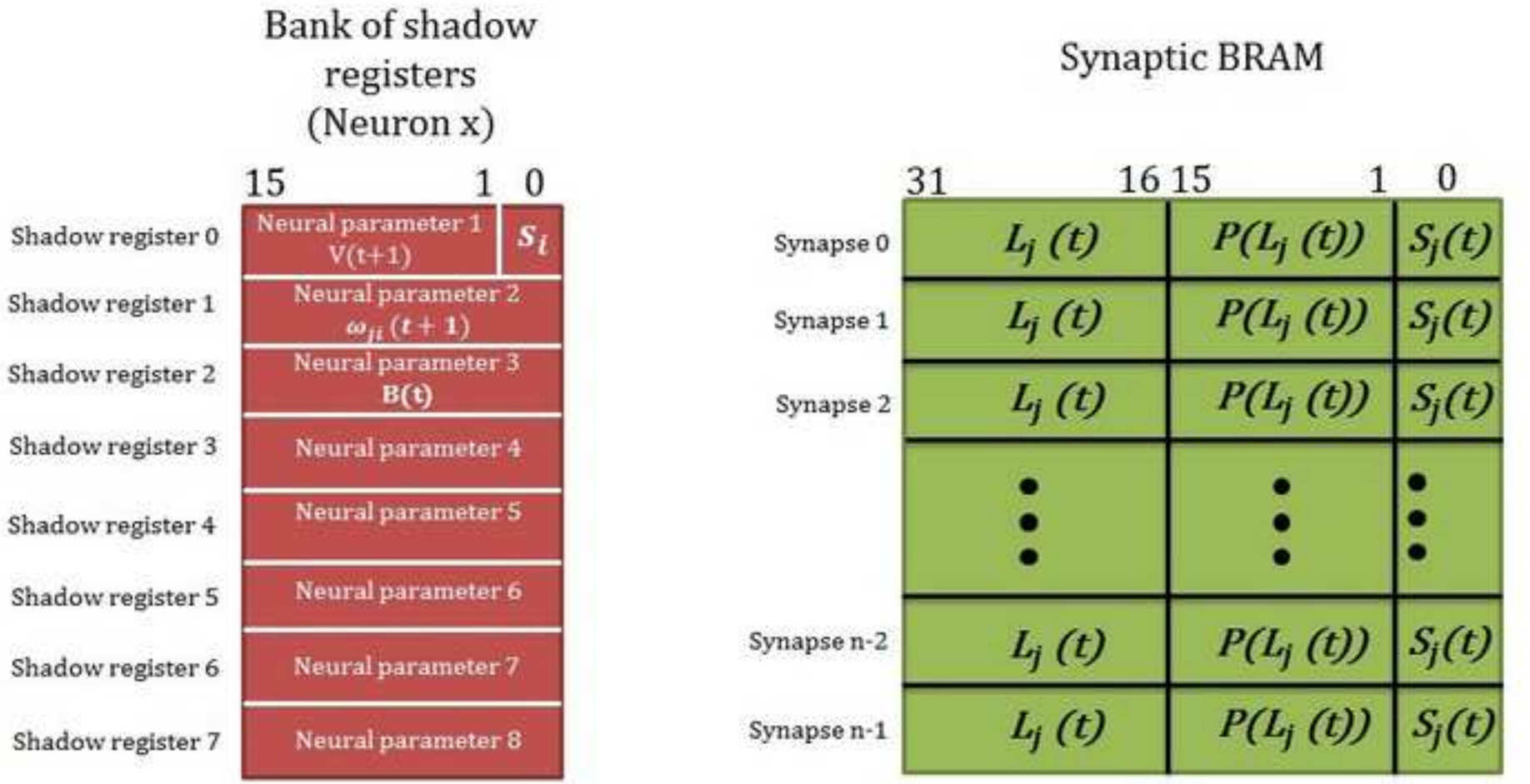


Figure
Click here to download high resolution image



*Biography of the author(s)
Click here to download Biography of the author(s): biography.pdf

LEGION-Based Image Segmentation by Means of Spiking Neural Networks Using Normalized Synaptic Weights Implemented on a Compact Scalable Neuromorphic Architecture

Authors:

Giovanny Sánchez received the M.S. degree at Instituto Politecnico Nacional, Mexico, in 2008, and the Ph.D. degree at Universitat Politecnica de Catalunya, Spain, in 2014. Currently, he is an Associate Professor in the Instituto Politecnico Nacional, Mexico. His main research interests include development of neuromorphic systems, encryption systems, image and signal processing applications.

Jordi Madrenas (Telecom. Eng. 86, Ph.D. 91, UPC, Barcelona, Catalunya, Spain) is currently Associate Professor at the Department of Electronic Engineering, Universitat Politècnica de Catalunya (UPC). He has participated in 5 European projects and coordinated 5 Spanish national research projects as well as several contracts with companies. At present, he coordinates a national project on MEMS-on-chip and microsensor bioinspired signal processing. He has co-authored more than 130 scientific journal and international conference papers, two books and 5 book chapters. Between 2000 and 2003 he was Vice-Dean of Studies of the Telecommunication Engineering School of Barcelona, UPC. His current research interests include analog, mixed-signal and digital VLSI and FPGA design CMOS-MEMS design and conditioning, ultra low-power design, bio-inspired/neuromorphic system implementation and rad-hard mixed-signal circuits.

Jordi Cosp received the M.S. and Ph.D. degrees (with honors) in telecommunication engineering from the Universitat Politecnica de Catalunya (UPC), Barcelona, Spain, in 1995 and 2002, respectively. From 1997 to 2000, he held a FI Research Fellowship in the Department of Electronic Engineering, UPC. In 2000, he became an Assistant Professor at UPC. He has participated in three European and national research projects in the last years. His research interests include neuromorphic engineering, analog and digital VLSI design, neural networks, nonlinear oscillators, and image processing implementation schemes.